



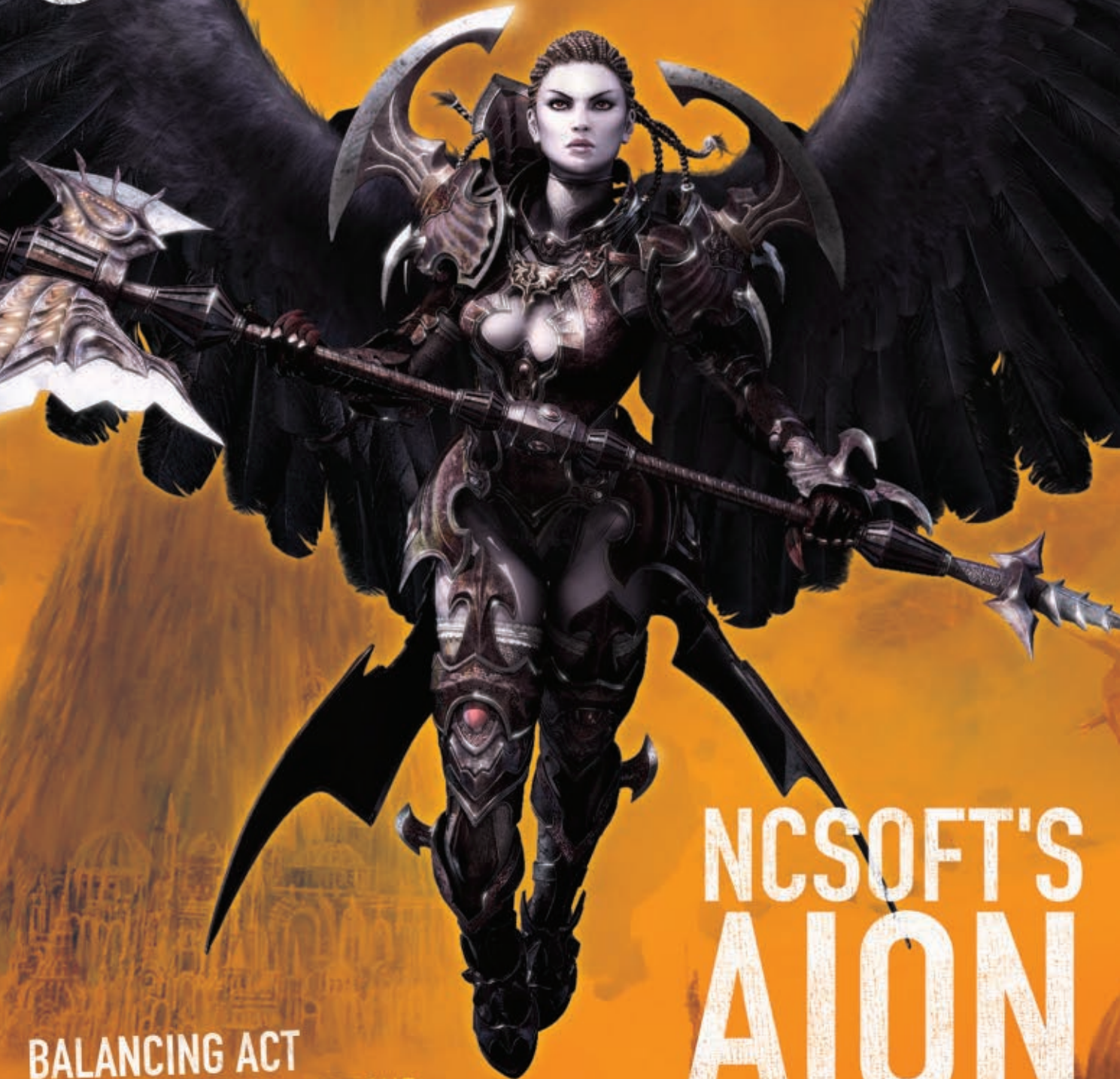
United Business Media

SPECIAL FEATURE: 2009 FRONT LINE AWARDS

VOL17 NO1 JANUARY 2010

# gamedeveloper

THE LEADING GAME INDUSTRY MAGAZINE



**BALANCING ACT**

USING GRAVITY TO IMPROVE  
CHARACTER ANIMATION

**NC SOFT'S  
AION**

POSTMORTEM

# Reliable Pathfinding Technology. Havok AI.



Physics Animation Behavior Cloth Destruction AI

**havok**<sup>™</sup>  
[www.havok.com](http://www.havok.com)

*Turning Creative Aspirations  
into Technical Realities<sup>™</sup>*

## POSTMORTEM

### 20 NCSOFT'S AION

AION is NCsoft's next big subscription MMORPG, originating from the company's home base in South Korea. In our first-ever Korean postmortem, the team discusses how AION survived worker fatigue, stock drops, and real money traders, providing budget and demographics information along the way.

*By NCsoft South Korean team*

## FEATURES

### 7 2009 FRONT LINE AWARDS

We're happy to present our 12th annual tools awards, representing the best in game industry software, across engines, middleware, production tools, audio tools, and beyond, as voted by the *Game Developer* audience.

*By Eric Arnold, Alex Bethke, Rachel Cordone, Sjoerd De Jong, Richard Jacques, Rodrigue Prallier, and Brian Thomas.*

### 15 RETHINKING USER INTERFACE

Thinking of making a game for multitouch-based platforms? This article offers a look at the UI considerations when moving to this sort of interface, including specific advice for touch offset, and more.

*By Brian Robbins*

### 27 CENTER OF MASS

This technical artist-oriented article from an ex pro gymnast shows how you can use math and physics (specifically dealing with gravity) to improve your character animation. The author most recently worked on UFC UNDISPUTED 2009, which has some of the best body interpolation and animation seen in games.

*By Eiko Oba*

## DEPARTMENTS

2 **GAME PLAN** *By Brandon Sheffield*  
Going Through the Motions

[EDITORIAL]

4 **HEADS UP DISPLAY**  
Open Source Space Games, new NES music engine, and Gamma IV contest announcement.

[NEWS]

34 **TOOL BOX** *By Chris DeLeon*  
Unity Technologies' Unity 2.6

[REVIEW]

38 **THE INNER PRODUCT** *By Jake Cannell*  
Brick by Brick

[PROGRAMMING]

42 **PIXEL PUSHER** *By Steve Theodore*  
Tilin'? Stylin'!

[ART]

46 **DESIGN OF THE TIMES** *By Damian Schubert*  
Get Real

[DESIGN]

48 **AURAL FIXATION** *By Jesse Harlin*  
Dethroned

[SOUND]

50 **GOOD JOB!**  
Konami sound team mass exodus, Kim Swift interview, and who went where.

[CAREER]

52 **EDUCATED PLAY**  
Focus on ENJMIN's student game AZ66

[EDUCATION]

56 **ARRESTED DEVELOPMENT** *By Matthew Wasteland*  
Dear Mr. Wasteland

[HUMOR]



# GOING THROUGH THE MOTIONS

## IS GESTURE CONTROL REALLY THE FUTURE?

### THE YEAR 2010 IS UPON US.

It should prove to be a time of iterative improvements, rather than major hardware shifts, and the area in which this is most apparent is peripherals. Within this year, or so it is said, there will be three home systems with motion or gesture control—the Wii of course, the PlayStation 3 with its motion wands, and the 360's Project Natal.

Everyone's chasing the motion and gesture train, after the success of the DS, the iPhone, and the Wii. But is motion really the reason these consoles are successful? Partially, sure—but that's not the whole story.

### YOU LOOK FAMILIAR

» Motion controls have been around for a long time. Light gun games in arcades and at home are primitive motion controllers, and they've been around since games began—one of Ralph Baer's first prototypes before the legendary Brown Box was a light gun that worked with a television. Touch controls, likewise, have been around for quite a while. The DS brought it to the masses, and the iPhone hammered it home, but PDAs have had touch control for years now, and have been host to games with major industry backing to boot (remember the Tapwave Zodiac?).

I would submit that aside from outstanding games like *Wii Sports* or *Boom Blox*, much of what's done on the Wii with motion control could be done with a normal controller, if a few design issues were solved. The thought struck me as I was playing *New Super Mario Bros. Wii*, tilting the controller to raise the end of a platform so that he could access a higher area—this could easily be done with shoulder buttons.

My point is not to trivialize the Wii Remote, but rather to point out that what Nintendo did when it released the DS and the Wii was not to revolutionize control. There were subtle upgrades certainly—the DS' two screens, and the Wii's

gyroscope and accelerometer were a step beyond what had been done previously, and the iPhone's multitouch interface hadn't really been done before with games in mind. The innovations were subtle—what was really disruptive, as Nintendo might say, was the marketing.

### YES, THESE ARE YOUR MOM'S VIDEOGAMES

» Nintendo told us that its systems were new, and disruptive—but what they told moms, grandmothers and grandfathers was that this device was fun for everyone. They could say that this was brand new, and just for them, and mostly be right in saying it. Nintendo took out ads in parenting and women's magazines, and blazed a trail of accessible television marketing that placed its consoles far away from the others, which were, at least in terms of marketing, very clearly for 17 year old boys in the early days.

Nintendo reminded people that it was the company that made *MARIO*, and they all remembered *MARIO*, right? That was probably the only game the target market had ever played that wasn't *TETRIS*. Nintendo knew the market it was going for, and targeted it perfectly. Parents, grandparents, and most importantly, families.

Now, Sony and Microsoft are releasing motion control expansions, both with the express intention of broadening their consoles' markets. But can they do it from where they are now? These companies don't have the benefit of a completely new launch with which to brand themselves, and have spent most of their consoles' lifetimes marketing to the hardcore. (Let's face it, Nintendo, aside from perhaps a brief stint in the 90s, never targeted the hardcore very directly, choosing to go after the youthful and light players with the bulk of its marketing bucks.)

Sony and Microsoft most likely have to rely on that 17–34

year old male to bring the console into the house (Sony may have it easier here, with its Blu-ray player, but the jury's still out on that), and market the peripherals as something they can plug into their existing system. Microsoft is rumored to be preparing a relaunch for the console with Natal, likely responding to the predicament I'm describing. Quite simply, my question is, no matter how nice the motion controls and cameras themselves may be, will these companies be able to rebrand themselves properly for the family set while continuing to push the blockbusters that have been their bread and butter? Right now, the 360's best selling game is *CALL OF DUTY MODERN WARFARE 2*. Far from a family game, that one, and a market the company is not likely to abandon.

### CATCH ME IF YOU CAN

» Sony and Microsoft have a long way to go before they can steal Nintendo's thunder. Sure, the 360 and PS3 have the edge in terms of game to hardware tie ratio, but that goes hand in hand with the hardcore. The Wii was purchased by a lot of people who only wanted *Wii Sports*, and maybe another Nintendo game every year thereafter.

My concern is that Sony and Microsoft may have a stigma to overcome before they can get the moms and grandmas involved in their console. They've both spent a lot of time promoting their machines as homes of blockbusters—and unlike the movie definition of that word, game blockbusters only appeal to a certain set of people.

2010 will be a very interesting year, with battles fought between Sony and Microsoft for dominance of the hardcore set, and between all parties for the "emerging market" set. As that demographic increasingly turns to social network games and the iPhone, this will be a battle that's hard won.

—Brandon Sheffield

Think Services, 600 Harrison St., 6th Fl.,  
San Francisco, CA 94107  
t: 415.947.6000 f: 415.947.6090

### SUBSCRIPTION SERVICES

FOR INFORMATION, ORDER QUESTIONS, AND ADDRESS CHANGES

t: 800.250.2429 f: 847.763.9606  
e: [gamedeveloper@halldata.com](mailto:gamedeveloper@halldata.com)

### EDITORIAL

#### PUBLISHER

Simon Carless | [scarless@gdmag.com](mailto:scarless@gdmag.com)

#### EDITOR-IN-CHIEF

Brandon Sheffield | [bsheffield@gdmag.com](mailto:bsheffield@gdmag.com)

#### PRODUCTION EDITOR

Jeffrey Fleming | [jffleming@gdmag.com](mailto:jffleming@gdmag.com)

#### ART DIRECTOR

Joseph Mitch | [jmitch@gdmag.com](mailto:jmitch@gdmag.com)

#### CONTRIBUTING EDITORS

Jesse Harlin

Steve Theodore

Daniel Nelson

Soren Johnson

Damion Schubert

#### ADVISORY BOARD

Hal Barwood Designer-at-Large

Mick West Independent

Brad Bulkley Neversoft

Clinton Keith Independent

Bijan Forutanpour Sony Online Entertainment

Mark DeLoura Independent

Carey Chico Pandemic Studios

### ADVERTISING SALES

#### GLOBAL SALES DIRECTOR

Aaron Murawski | [amurawski@think-services.com](mailto:amurawski@think-services.com)

t: 415.947.6227

#### MEDIA ACCOUNT MANAGER

John Malik Watson | [jmwatson@think-services.com](mailto:jmwatson@think-services.com)

t: 415.947.6224

#### GLOBAL ACCOUNT MANAGER, EDUCATION AND RECRUITMENT

Gina Gross | [ggross@think-services.com](mailto:ggross@think-services.com)

t: 415.947.6241

#### COORDINATOR, EDUCATION AND RECRUITMENT

Rafael Vallin | [rvallin@think-services.com](mailto:rvallin@think-services.com)

t: 415.947.6223

### ADVERTISING PRODUCTION

#### PRODUCTION MANAGER

Robert Steigleider | [rsteigleider@ubm-us.com](mailto:rsteigleider@ubm-us.com)

### REPRINTS

#### WRIGHT'S REPRINTS

Ryan Pratt | [rpratt@wrightsreprints.com](mailto:rpratt@wrightsreprints.com)

t: 877.652.5295

### THINK SERVICES

CEO THINK SERVICES Philip Chapnick

GROUP DIRECTOR Kathy Schoback

CREATIVE DIRECTOR Cliff Scorso

CHIEF INFORMATION OFFICER Anthony Adams

### AUDIENCE DEVELOPMENT

TYSON ASSOCIATES Elaine Tyson

e: [tysonassoc@aol.com](mailto:tysonassoc@aol.com)

LIST RENTAL Merit Direct LLC t: 914.368.1000

### MARKETING

MARKETING SPECIALIST Mellisa Andrade

e: [mandrade@think-services.com](mailto:mandrade@think-services.com)

### UBM TECHNOLOGY MANAGEMENT

CHIEF EXECUTIVE OFFICER David Levin

CHIEF OPERATING OFFICER Scott Mozarsky

CHIEF FINANCIAL OFFICER David Wein

CORPORATE SENIOR VP SALES Anne Marie Miller

SENIOR VP, STRATEGIC DEV. AND BUSINESS ADMIN. Pat Nohilly

SENIOR VP, MANUFACTURING Marie Myers



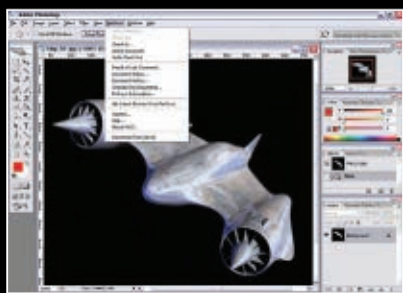


## Introducing P4GT, a productivity feature of Perforce SCM.

The Perforce Plug-in for Graphical Tools, P4GT, makes version control painless by seamlessly integrating Perforce with leading graphical tools. Drop-down menus allow access to Perforce from within 3ds Max, Maya, Softimage XSI, and Adobe Photoshop.

Art and development teams can standardize on Perforce to version and manage both source code and digital assets. Enhanced collaboration during the design process helps teams to work together in real time to release small patches or create whole new worlds.

P4GT is just one of the many productivity tools that comes with the Perforce SCM System.



P4GT

**PERFORCE**  
SOFTWARE

**Download a free copy of Perforce, no questions asked, from [www.perforce.com](http://www.perforce.com). Free technical support is available throughout your evaluation.**



## 2010 space is the place

SPACE GAMES MAY HAVE FALLEN OFF THE COMMERCIAL RADAR IN RECENT YEARS (WITH THE NOTABLE EXCEPTION OF EVE ONLINE) BUT INDEPENDENT DEVELOPERS HAVE KEPT THE FUSION DRIVES LIT, ILLUMINATING THE INTERSTELLAR REACHES WITH AN ARRAY OF OPEN SOURCE AND COLLABORATIVE PROJECTS. HERE'S A SAMPLING OF WHAT'S OUT THERE:

### ELITE

[www.iancgbell.clara.net/elite/index.htm](http://www.iancgbell.clara.net/elite/index.htm)



David Brabin and Ian Bell created the archetypal space sim ELITE for the BBC Micro

back in 1984. A completely open-ended experience, ELITE was a true sandbox that allowed players to explore its procedurally generated wire-frame space, eventually fighting and trading their way to an "Elite" ranking. In the years since, the game has been ported to a wide range of computer systems and Ian Bell has made ROM versions of ELITE freely available for a number of them, including the Nintendo Entertainment System.



**THE UR-QUAN MASTERS**  
<http://sc2.sourceforge.net>

Toys For Bob's STAR CONTROL 2 has remained a classic of the genre since its release in 1992. Combining open-world trading, exploration, and combat with a deep and cleverly-written dialog system, STAR CONTROL 2's vast universe was given extra charm by a quirky graphic design that included art and music contributions from old school TSR illustrator Erol Otus.

Fortunately STAR CONTROL 2's refined game play can

be enjoyed today thanks to Toys For Bob's release of the 3DO version's source code. Now integrated into an open source project called THE UR-QUAN MASTERS, the game runs on OS X, Windows, and Linux. THE UR-QUAN MASTERS adds a number of new features including add-on remix music packs from the Precursors collective ([www.medievalfuture.com/precursors](http://www.medievalfuture.com/precursors)) and networked multiplayer combat.



**CELESTIA**  
[www.shatters.net/celestia](http://www.shatters.net/celestia)

Not exactly a game but equally compelling, CELESTIA is an open source space simulation for OS X, Windows, and Linux that attempts to recreate the known galaxy and beyond. Incorporating the latest data from space research, CELESTIA allows the user to explore the tremendous scale of the universe, from individual satellites measuring only a few feet across, to star clusters that cover light years of distance. Using simple, space-ship like controls, the user can set out in any direction and along the way discover just how the incredibly big our galaxy is. Its source code is freely available and a central repository for user-created add-ons for CELESTIA has

been established at The Celestia Motherlode ([www.celestiamotherlode.net](http://www.celestiamotherlode.net)).



**OOLITE**  
[www.oolite.org](http://www.oolite.org)

Bringing the ELITE experience into the 21st century, OOLITE (OBJECT ORIENTED [E]LITE) is an open source project that closely follows the spirit of ELITE with the addition of OpenGL graphics for Mac OS X, Windows, and Linux. The game also supports a vast selection of community built mods through its OOLITE eXpansion Packs.



**VEGA STRIKE**  
<http://vegastrike.sourceforge.net>

VEGA STRIKE is an open source project that aims to create an expansive 3D space simulation. Very much in the style of ELITE, VEGA STRIKE provides a dynamic universe for players to freely explore, trade, and fight across. The project also forms the basis of a number of conversions including a remake of WING COMMANDER: PRIVATEER. Currently VEGA STRIKE is in a

stable version 0.5.0 for OS X, Windows, and Linux.



**NOCTIS IV**  
<http://anywherebb.com/bb/layout/html/frameset.html>

NOCTIS (now in its fourth iteration) is an ambitious attempt at creating an entire procedurally generated galaxy for players to explore. Explicitly not a simulation of the known universe, the vast space of NOCTIS is completely unique to the game and players have mapped only a portion of its true dimensions. Playing NOCTIS is a quiet, solitary experience, devoid of the combat and commerce that are the mainstays of other space sims. Instead, players drift through the game's cold reaches, logging their discoveries into a shared database.

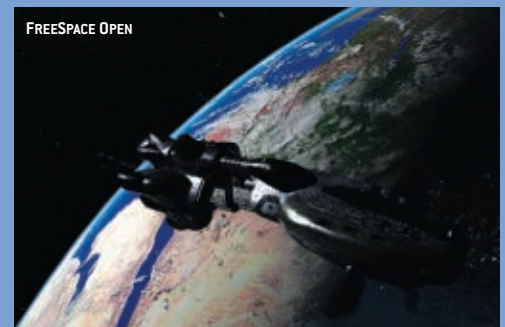
NOCTIS is the creation of Alessandro Ghignola and he has made the source code freely available although it is under a limited license that makes changes and redistributions subject to his

approval. However, the NOCTIS community has created a number of essential add-on including NoctisMapper to facilitate exploring the game.

**Freespace Open**  
<http://scp.indiegames.us>

Volition Inc's FREESPACE 2 is a fondly remembered warfare simulation that featured lithe corvettes and fighters slugging it out against massive capital ships in deep space. Although it was published a decade ago, players can continue to enjoy the game on modern hardware. Released as open source code in 2002, the FREESPACE 2 engine is now the basis of the FreeSpace Open project for OS X, Windows, and Linux, which continues to polish and improve what was already a graphic showcase. Although playing FREESPACE 2 with the upgraded engine still requires you to own a legal copy of the original software, enterprising coders are utilizing the engine to create standalone games as well. A few of the more ambitious standalone efforts include THE BABYLON PROJECT (<http://babylon.hard-light.net>), and WING COMMANDER SAGA ([www.wcsaga.com](http://www.wcsaga.com)).

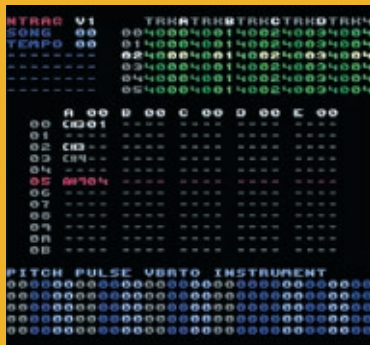
—Jeffrey Fleming



# free nes audio engine nijuu released

**NEIL BALDWIN WAS AN NES CHIPTUNE** maker back in the old days, adding his scores to Eurocom's games, where he was a cofounder, and still works today. Most recently, he's created and released Nijuu, a free audio engine for NES music creation, using only the original 4 channels, "so no DPCM samples," he says. "It does have some pretty sophisticated features though, such as automated single-voice echo, voice multiplexing and a 'wavetable' virtual drum track."

Nijuu is available now at the official website (<http://duty-cycle-generator.com/nijuu/nijuu.html>). To further describe the



software, "It's not a tracker and it's not really MML [music macro language]," says Baldwin. "You do have to type [or copy and paste] the music into text files and compile the binary data using a 6502 assembler. It can output either binary data that you can include in your own project/ROM, or .NSF files if you just want to play back the music in a NES audio player."

Baldwin was inspired by hearing some of his old game tracks on YouTube, and subsequently discovering the thriving modern chiptune scene. "I read a lot about composers using MML," he says, "and while I could see the benefits [and similarity to my own old NES audio working method], it left me wanting some of the features of my old audio engine, most specifically the use of multiplexing/virtual channels."

Baldwin's music was known for bringing a very C64-like sweeping electronic quality to the NES, making for some of the best Western chip music of the era. "This in turn got me thinking about how I might've

approached writing a NES audio engine," added Baldwin, "free from the constraints of existing within a game [for example no CPU use restrictions and no RAM restrictions]. So I set about trying some of the crazier ideas and the foundations of Nijuu were conceived."

Baldwin has provided us with a mini-postmortem of the audio engine's creation: "After going through several different 6502 assemblers I settled on one called 'ASM6,' written by a guy who goes by the name of 'loopy.' Using this in combination with the emulator Nestopia, I had a quick development method through which to prototype all the mad ideas I had for Nijuu. Amazingly, you can get tons of NES development tools for free, and there's a wealth of information for developers on sites such as nesdev."

"Many of the structural ideas for Nijuu were taken directly from my old NES audio engine, but I added so many features and effects that it now shares very little similarity with that old code."

"I didn't start creating it with any real goal in mind, certainly not to release it to the public. I just set out to prove several ideas I'd had from when I was creating NES music in the old days. The development process was pretty much "Wouldn't it be cool if Nijuu could do X, Y, and Z!?" and then I'd figure out a way to do it. Some were easier than others—the single-voice echo effect was the most difficult to solve. The actual code is relatively small and simple but it took me quite a lot of head-scratching to figure out a [usable] way to achieve it."

"At some point I recognized that what I was creating was unique, and might be of interest to other NES composers—so I began to try to shape it into something that other people could use."

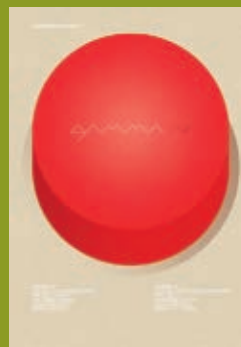
Nijuu is a work in progress, but is currently available in beta form for interested musicians. Baldwin is also in the process of creating a NES tracker called NRTQ—find more information, as well as selected soundtracks to his NES titles, on his site: <http://duty-cycle-generator.com>.

—Brandon Sheffield

# gamma announces one-button game competition for gdc

**MONTREAL'S KOKOROMI COLLECTIVE HAS ANNOUNCED** its theme for the Gamma 4 competition, with game makers challenged to make "innovative, experimental new games played with just one button." Kokoromi is partnering with Think Services' Game Developers Conference to bring the fourth edition of its renowned Gamma game showcase to GDC 2010 next March.

Comparable to a longer-form, targeted version of the "indie game jam" concept, previous years' themes have included Gamma 01: Audio Feed (games driven by live audio), gamma 256 (games with extremely small pixel dimensions), and GAMMA 3D (games using red-blue stereoscopic 3D). Standout games like PASSAGE, PAPER MOON,



and SUPER HYPERCUBE resulted.

This time around, the Gamma organizers have framed the competition as follows: "Gestural controls, multi-touch surfaces, musical instruments, voice recognition—even

brain control. Games are moving beyond the iconic hand-held controller, and into the future. But is the secret to good games found in high-tech interface hardware? Kokoromi proposes that game developers can still find beauty in absolute simplicity."

Taking place on the evening of Wednesday March 10th, 2010, the Gamma 4 kickoff event bridges the end of the Independent Games Summit and the start of the main GDC. The playable games will be revealed at the Mezzanine, a venue housed in a historic two-story warehouse near Mint Plaza, in the heart of San Francisco's SoMa district. The curated games will be featured on large projections, and accompanied by live DJs.

Game creators who are selected for presentation at Gamma 4 will be awarded GDC 2010 All-Access passes, and developers around the world now have from December 1st 2009 until January 31st 2010 at midnight Pacific time to complete and submit their single-input creations. Full submission rules and guidelines are available at [www.kokoromi.org/gamma4](http://www.kokoromi.org/gamma4).

—Staff

Register before **FEBRUARY 4<sup>TH</sup>** and **SAVE UP TO 35%!**



# GDC 10

**Game Developers Conference<sup>®</sup>**

**March 9-13, 2010** | Moscone Center, San Francisco

**[www.GDConf.com](http://www.GDConf.com)**



**Learn. Network. Inspire.**



# FRONT LINE AWARDS



**GAME DEVELOPER MAGAZINE'S MISSION FOR OVER FIFTEEN YEARS HAS BEEN TO PROVIDE GAME CREATORS WITH INFORMATION, NEWS, AND** articles that pertain directly to them. The Front Line Awards are an official way of recognizing one specific aspect of the industry: the tools that developers need to do their jobs.

Each year, we look at the lineup of new products and new releases of important tools in the categories of game engines, books, art tools, middleware, audio tools, and programming/production tools. The resulting Front Line Award winners represent the most innovative, user-friendly, and useful products from behind the scenes of the world's best video games. In addition to the Front Line Award winners we also want to pay special tribute with our Hall of Fame Award to a product that has made a lasting impact on the game industry year after year (and which was not eligible to win in its specific category as a result).

Nominations for this year's Front Line Awards were open to all new software products and new versions of software products related to game development released between September 1, 2008 and August 31, 2009. In determining the winners of the 2009 awards we went through a multistep process. Open nominations were held in October, and from that list we consulted with our advisory board to narrow down the results to five entries in each category. We then handed the finalists over to you, the readers of *Game Developer*, via an invitational online survey in November, so that you could have a voice in picking the recipients of the Front Line Awards. After tabulating the survey responses we are proud to present the winners along with commentary by developers from throughout the game industry. A special thanks goes out to everyone who contributed to this year's Front Line Awards and congratulations to all the finalists and winners.

—Jeffrey Fleming

★ HALL OF FAME ★

# BINK VIDEO

RAD GAME TOOLS ★ WWW.RADGAMETOOLS.COM

Game development was once described to me as building a house on top of a moving train. In space. With dinosaurs. It's never an easy process, and you have to learn to deal with the unexpected. Storytelling in games is no different.

huge, expansive areas, tell a rich and fleshed out story, and still fit within memory and performance constraints. Using Bink let us solve those challenges. Bink's strong integration with Epic's Unreal Engine 3.0 meant that we could use the technology

Bink's integration with the Unreal Engine came in. By bringing the movies into the engine as texture movies, we could manipulate them with the material shader system. We were able to key out colors, add distortion and alpha masks, with just

their original size, without the players noticing a difference. This allowed us to stream the Binks off the disc, and reinvest that memory overhead back into the game world.

Our use of Bink doesn't stop here—it extends

With a little help from our engineering department, we decided to turn the introduction into a Bink, and strive for a continuous experience from the moment the player starts a game to the time they gain control of their character. Unreal allowed us to do a frame-by-frame capture of our intro, while Bink allowed us to take the result and play it back in high definition with no artifacting. Both technologies worked seamlessly together, and the result was a taking a great intro concept and turning it into a great experience for the user.

Bink didn't stop there, though. With another assist from our engineers, we were able to turn Bink's support of multi-track audio into our one-stop solution for localization. We did this by separating one track for effects and music, and using the remaining tracks to store dialogue tracks for each region we planned to ship in. Instead of having to manage five different videos, we could use one single Bink video to do the work of all of them.

As storytellers, our goal is to keep the player engaged and entertained, while reminding them as little as possible that they're playing a piece of software. Achieving that goal is a different process for each game, and it requires flexible solutions. That's why we use Bink.

— Brian Thomas, Gearbox Software

Gearbox Software's BORDERLANDS.



Whether it's a last minute re-write, cutting a level, or bending the game engine beyond its imagined limits, there's always a new challenge to conquer. That's what makes flexibility such a comforting thing during development. That's why Gearbox chooses to use Bink. When developing our latest title, BORDERLANDS, we had some very difficult hurdles to clear. We needed to create

straight out of the box, and hit the ground running. The largest technical challenge we faced was integrating our female narrator, the Angel, into our UI. We needed a way for her to hack your communication system to deliver her messages. We chose to do this with some highly stylized FMVs, but the effect we were looking for wasn't quite there. This is where

a few clicks. The resulting material could easily be used in UI, or anywhere else in game. But the work wasn't quite done. We now needed a way to do this without blowing our memory constraints. Bink's toolset allowed us to easily scale, compress, and tweak our FMVs until we found that magical spot where the resulting videos were a fraction of

deep into the game. Our introduction cinematic was created in-engine, and was originally intended to be played that way. As the game neared the end of development, we began to understand that the user's experience was being marred by watching too many loading screens as the game transitioned between the menu, the intro, and the game.

## ★ ART TOOL ★

## AUTODESK 3DS MAX 2010

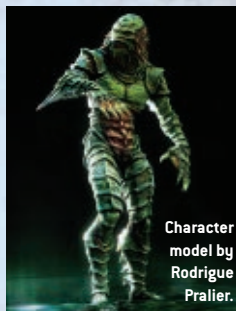
AUTODESK ★ [HTTP://USA.AUTODESK.COM](http://usa.autodesk.com)

I have been a 3ds Max user at home and at work for five years, and I employ it in all my high-res modeling for video game or high-res scenes. I have also been a big user of mental ray for my renders, and 3ds Max's mental ray integration is very easy to use, with preset settings for global illumination's Final Gather with draft, low, medium, high, and very high settings. All these features are still there but with 3ds Max 2010 we have even more like soft shadow Precision, glossy reflect, and refract Precision,

and they are just as easy to use with their sliders.

I often use its cloth simulation to quickly generate a rough pass on the cloth of some of my characters. I use it for capes, pants, and scarves, and 3ds Max 2010 allows me to work with more polygons on screen and to generate nice cloth effects and hair quicker than before.

I really like that when I do my projections, if I put a bump map or a normal map on my source model the projection will take all this bump information and bake it onto a normal map



Character model by Rodrigue Pralier.

on my low-res model. It's a very quick way to have nice details and it allows me to keep my 3ds Max scene very light with fast render times and more details in my occlusion passes.

In video game production you need to be fast and clean with optimization; in 3ds Max 2010 the Pro-optimizer and the Multi-res tools make that job very easy. I have also been a big Polyboost user, and to see that most of these tools are now part of the default toolset of 3ds Max 2010 is just great. The new Graphite modeling tool set has all the tools that modelers need and the smoothing display of your objects can be changed in one click.

Among these tools, the ones I use the most are the loop and ring modes and the dot

ring tools. The flow connect is also very useful for high-res modeling. But my favorite new tool is the swift loop, which allows me to create new geometry in one click, and gives me the ability to place it where I want in a click. Another nice feature of the 3ds Max 2010 viewports is the ability to display soft shadows and ambient occlusion on your models in real time.

I was very happy with the 2009 version but 3ds Max 2010 is a pleasant surprise that has made my work flow even faster.

—Rodrigue Pralier, EA Montreal

## ART TOOL FINALISTS

- ★ Corel Painter 11
- ★ Autodesk Mudbox 2009
- ★ Luxology modo 401
- ★ Pixologic ZBrush 3.12b

## ★ AUDIO TOOL ★

## PRO TOOLS 8

DIGIDESIGN ★ [WWW.DIGIDESIGN.COM](http://www.digidesign.com)

Beginning life as a stereo recording and editing system way back in 1989, Digidesign's Pro Tools Digital Audio Workstation (DAW) now packs some serious punch by offering up to 192 audio tracks, 128 virtual instrument tracks, a fully functioning MIDI sequencer and score editor, as well as high resolution recording up to 192kHz across 160 channels of I/O.

In its latest incarnation Pro Tools 8 is available on both Mac and PC platforms, which opens the doors to many PC-only developers and offers extensive file output options, suitable for all in-house or middleware audio tools across all platforms.

During the game development cycle, from planning to final implementation, the Pro Tools platform has become an essential part in the creation of audio assets, be it from a simple dialogue recording right up to a fully interactive music score. The software is packed so deep with features, that one often asks the question "how do you use yours?"

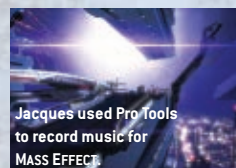
When recording dialogue for your title, you will find Pro Tools at the heart of any professional voice over studio, and with such a great array of dialogue-specific plug ins from the likes of Celemony Software (Melodyne), Mu Technologies (Mu Voice), and Synchro Arts (VocAlign Pro)

among others, it is incredibly quick and easy to record, edit, and process thousands of lines of dialogue, all named and time stamped and converted to your desired format.

Sound designers often utilize third party sound library applications such as Sound Miner, and having drag and drop functionality can quickly get their effects into the Edit window to begin designing or editing—whether designing cinematic sequences via Pro Tools' integrated movie playback, or creating sounds for implementation directly into the game.

For composers, there is nothing simpler than leaving the scoring stage after a large

orchestral recording and loading up the Pro Tools session for mixing, mastering, and implementation, either at another facility or the developers own studio. Indeed, many developers now ask for Pro Tools sessions as the final



deliverables, giving them even further flexibility for final tweaking and last minute implementation. The "Import Session Data" function is also one of the greatest time saving features of any audio

platform, allowing the user to import various data and settings, tracks, channel strips, audio and video regions from any Pro Tools session file, even if they were created with older versions of the Pro Tools software. This enables users to swap data from any corner of the world, and with remote recording functionality from the likes of Source Connect, the virtual recording session is indeed alive and well.

And even though all the above features and potential usages barely scratch the surface, Digidesign really has put the "pro" in Pro Tools. How do you use yours?

—Richard Jacques  
[www.richardjacques.com](http://www.richardjacques.com)

## AUDIO TOOL FINALISTS

- ★ FMOD Designer 4.27, Firelight Technologies
- ★ Miles Sound System 7.2e, RAD Game Tools
- ★ Vivox Precision Studio SDK v. 3.0.6, Vivox
- ★ Wwise 2009.2, Audiokinetic

★ MIDDLEWARE ★

Volition Inc's RED FACTION: GUERRILLA.



# HAVOK PHYSICS

HAVOK ★ [WWW.HAVOK.COM](http://WWW.HAVOK.COM)

Simply put, RED FACTION: GUERRILLA would have been stillborn in the concept phase without Havok. What started out as a designer's pipedream became a living, breathing, crumbling, and violently exploding reality thanks to the power and flexibility that Havok brought to the table. But it is not the extensive, constantly evolving code base and feature set that won it this award for the second year in a row; it is the top-notch support that comes along with it.

That is not to say that

the engine itself is less than impressive, far from it. Havok Physics' strength and stability allowed us to build a custom destruction engine on top of its base that can handle everything from a simple wooden crate to an eight-story skyscraper without missing a beat. The extensive documentation provides a wealth of knowledge about fine-tuning all the knobs and switches to keep the frame rate steady even when the sky is literally falling. Its cross platform support meant that what took years to develop

on the Xbox 360 only took weeks to port over to the PS3 with full SPU support for our destruction engine.

The real reason Havok deserves this award is its stellar support staff. Even if the documentation is perfect there will always be issues that crop up that require human interaction. Whether you find a bug in low-level code or are trying to figure out the best way to accomplish a given task, help is only a short email away. On the off-chance that there is a bug in internal code they are

happy to deliver a custom build for you. Not in the next release or patch—now. And now is the best time to have what you need.

A key factor is that each product is assigned a caseworker that gets to know you and your game. You don't have to waste time on each new ticket bringing a new guy up to speed on the structure of your game before diving in to the problem at hand. It may not sound important now, but when you are moments from shipping there is no better feeling than knowing you

have the ability to call up a friend on the inside who knows exactly what is going on. Whether it is through email, on the phone, or on-site, the Havok support engineers are a pleasure to work with.

So thank you Havok, for letting us stand on your broad shoulders and build amazing games that were the stuff of dreams a few short years ago. Keep doing what you do so well and we will keep finding ways to use Physics to make those dreams come true.

—Eric Arnold, Volition, Inc.

MIDDLEWARE FINALISTS

- ★ NaturalMotion morpheme 2.0
- ★ Scaleform Gfx 3.0
- ★ Havok Behavior 6.6.0
- ★ Interactive Data Visualization SpeedTree 5.0

## ★ ENGINE ★

## UNREAL ENGINE 3

EPIC GAMES ★ [WWW.UNREALTECHNOLOGY.COM](http://WWW.UNREALTECHNOLOGY.COM)

I started using Unreal Technology in the late '90s. With every year that passed I saw the engine and its accompanying tools improve in giant leaps, giving developers ever more power and possibilities.

Over the years I have also worked with various in-house engines. While the games we produced with those engines were awesome, we always lost time during production because of the limited capabilities and weaknesses of the engine used. I have seen problems seriously hamper production many times,

things that I know would have been simple using Unreal Engine 3. During my career I have worked on five Unreal Engine games, and the speed at which we were able to develop those games was mind blowing.

Can you make an awesome game with pretty much any engine? Of course, but at what cost? Is reinventing the wheel in today's competitive games industry sane?

One of Unreal Engine's strengths is that it allows you to focus on the game rather than the technology. I always look at game development

as a three-step process. The first step is a solid idea, the second step is making the idea technically feasible as a game, and the third step is making the game fun to play. By using Unreal Engine 3 you get to skip that second step. You get to take your idea, and turn it into something fun right away, something that just works.

I especially value the power Unreal Engine 3 gives me, as an artist and designer. Features like the visual scripting tool Kismet and the Material Editor allow me to prototype a gameplay

idea or a graphical effect in a very short amount of time, and iterate fast. I have never seen another game engine come even close to that much power and speed!

Unreal Engine 3 is not only a very powerful engine and toolset; it also has a strong community backing it. There is a big pool of experienced Unreal Developers to tap into, allowing you to jumpstart your production, removing the need to intensively train new hires on in-house technology. The amount of info and support that can be found via both

the official support channels and throughout the wider Unreal developer community is immense.

In short, Unreal Engine 3 and its tools are not only the result of many years of development, but also years of polish and fine tuning, years of testing in big productions, and perhaps most importantly, years of feedback. The result is an unrivalled piece of technology, which is being constantly refined and improved.

— Sjoerd De Jong,  
Toltec Studios

## ENGINE FINALISTS

- ★ Crytek CryEngine 3
- ★ Emergent Game Technologies Gamebryo LightSpeed
- ★ Valve Source 1.6.1.6
- ★ Unity Technologies Unity 2.5.1



Toltec Studio's THE BALL.

★ PROGRAMMING TOOL ★

# ADOBE FLASH CS4

ADOBE ★ [WWW.ADOBE.COM/PRODUCTS/FLASH](http://WWW.ADOBE.COM/PRODUCTS/FLASH)

When I was approached to write an article awarding Flash CS4 the Best Programming Tool award I simply said to myself: Of course it won, Flash is awesome! Having worked as a Flash game developer for the last six years I definitely recognize Flash as an invaluable tool in the industry and Adobe's latest work on Flash CS4 continues to reinforce that sentiment.

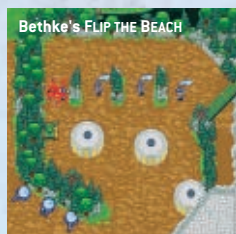
Flash CS4 has taken many of the strengths of the previous versions and enhanced them to provide a more robust and user-

friendly tool by doing things as simple as adding sliders for most value setting inputs or adding advanced new features such as Bone and 3D Rotation tools.

One of the reasons that Flash is such a great tool for game development in general is the ease with which users can get started and get results. A beginner can easily jump into the IDE and almost immediately begin animating and coding, while a veteran has a robust feature set that allows them to do both small and large-scale projects. The ability to

quickly take an idea from concept to completion is one of Flash's greatest strengths, and makes working with this tool a very rewarding experience because you can almost immediately see the results of your efforts.

There's a lot of focus in CS4 on the new animation and content related features, but this should in no way detract from the programming capabilities that this tool offers. Admittedly, when compared to other products like Adobe's Flex Builder 3, the Flash code authoring environment definitely has



some room for improvement but unlike Flex Builder, I think it has achieved a special balance between content and code creation that really makes it stand apart as the definitive all-in-one tool for simpler game development.

The bottom line is that what Flash does, it does

extremely well and for anyone considering making their first game or seasoned veterans looking for new tools, I would highly recommend Flash CS4. The blending of an animation suite and programming tool that CS4 provides continues to build upon the solid reputation of both Adobe and the Flash product and make it a clear winner for the programming tool award.

—Alex Bethke

PROGRAMMING/PRODUCTION TOOL FINALISTS

- ★ Hansoft 5.3
- ★ Perforce 2009.1
- ★ SlickEdit 2009 ver 14.0
- ★ Microsoft XNA Game Studio 3.1

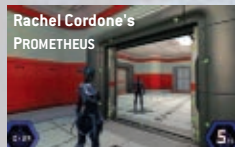
★ BOOK ★

# MASTERING UNREAL TECHNOLOGY: VOL. 1

BY JASON BUSBY, ZAK PARRISH, AND JEFF WILSON

SAMS PUBLISHING ★ [WWW.INFORMIT.COM/SAMS](http://WWW.INFORMIT.COM/SAMS)

The downside to being able to make anything you want is figuring out where to start. With an engine as powerful as Unreal, opening the editor



for the first time can be very intimidating. Luckily for anyone just getting into the game development scene, 3D Buzz continues its Unreal series of books with *Mastering Unreal Technology Volume 1* for the

Unreal Engine 3. I really wish resources like this were available when I first started!

3D Buzz has been in the Unreal scene for over five years now and has produced an incredible amount of video training guides for the Unreal Engine, including an entire DVD's worth in the UNREAL TOURNAMENT 2004 SPECIAL EDITION and again in the UNREAL TOURNAMENT 3 COLLECTOR'S EDITION. Needless to say, they know Unreal.

The book starts out with a quick history of Unreal and the engine, and then gets

right into the good stuff. Most people would completely overlook chapter 2, "Overview of Game Development," but I'm glad that the folks at Buzz knew the importance of putting this before any actual tutorials. Too many people come into the community with sky high ideas without realizing that game development is a lot of hard work, and sometimes gets monotonous, and yes, downright boring. They release a few concepts, make no further progress, get frustrated and quit. Anyone reading this book should

stop to think about why this chapter was included. In my opinion, it's the most important chapter.

The rest of the book gets into the nitty gritty of UnrealEd, and there is a lot of material to cover. From basic geometry construction, lighting, materials, and terrain—everything is covered not just with simple descriptions but also with step-by-step examples. The Kismet chapter in particular goes into much more detail and creates a more complicated example than I

would have expected, yet by guiding the reader through the process slowly, it never becomes overwhelming.

And to drive home the point that this is the only resource you'll need, you get a free copy of UNREAL TOURNAMENT 3 with the book. Bonus!

I'd like to congratulate 3D Buzz for winning the Front Line Awards. I'm looking forward to seeing how they go about teaching UnrealScript in their upcoming Volume 3!

—Rachel Cordone,  
[www.anglemapper.com](http://www.anglemapper.com)

BOOK FINALISTS

- ★ *Game Coding Complete 3rd Edition* by Mike McShaffry, Charles River Media
- ★ *Game Engine Architecture* by Jason Gregory, AK Peters
- ★ *Racing the Beam: The Atari Video Computer System* by Nick Montfort and Ian Bogost, The MIT Press
- ★ *Real Time Cameras: A Guide for Game Designers and Developers* by Mark Haigh-Hutchinson, Morgan Kaufmann

**EXPERIENCE THE POWER  
OF EXTREME 3D**

Autodesk® 3ds Max® 2010 gives you fast access to a wide range of tools for 3D professionals looking to shape new worlds, manage complexity and evolve their workflows.

Learn more at [autodesk.com/3dsmax](http://autodesk.com/3dsmax)

**Autodesk 3ds Max 2010**



EVEN IF  
YOUR COMPANY  
BUDGET EQUALS  
YOUR CREDIT  
CARD LIMIT,  
**your game can be Unreal.**



No matter what size your budget. No matter what type of game.  
Unreal can be your game engine. Email Mark Rein at [getunreal@epicgames.com](mailto:getunreal@epicgames.com).



# RETHINKING USER INTERFACE

## UI IN THE LAND OF THE TOUCH SCREEN

**GAME DEVELOPERS HOPING TO CREATE GAMES FOR THE LATEST TACTILE-ORIENTED PLATFORMS** today are facing an incredible challenge. The best of these developers will turn this challenge into an opportunity to innovate at a level not seen in the industry for years. The fact is touch input and accelerometer controls are becoming commonplace for many platforms, and game developers can ignore this trend only at their own peril.

While the success of the iPhone and iPod Touch has resulted in a viable market for game developers, it is not the only platform incorporating touch, multi-touch and accelerometer input. Nearly all next-gen mobile devices use some combination of these technologies, including Google Android-based devices, the Zune HD, Palm Pre and Pixi, and even the latest BlackBerry smartphones. Looking ahead, it is highly likely that over the next couple years the majority of higher-end mobile devices will use some or all of these features. The impact goes beyond just mobile applications though—multi-touch support is built into the core of Windows 7 and Apple's Magic Mouse is a multi-touch device. Users will soon start to see and expect this type of interface on their primary PCs and not just their cell phones.

This article covers several specific things that developers need to be concerned about when developing around these capabilities. While some may seem superficially obvious, most require careful attention and subtle rethinking in terms of how players experience and interact with the game, and what changes developers need to make in order to better enable ease of use for players.

## DUDE, WHERE'S MY SCREEN?

» Visibility of the screen becomes a huge issue when working with touch screen and accelerometer input. A player's hands and fingers will constantly be covering parts of the screen, and the simple act of controlling the game via accelerometers may make it hard to see what's happening.

As a general rule if a player is touching something then they will be unable to see anything below that point and angling out toward the edges of the screen. Poorly-designed controls and UI placement can easily result in the player's hands blocking out large portions of the visible area while playing the game. Fortunately this is a very easy fix for most game UI, as the general solution is to flip element placement vertically.

Instead of placing drop down menus and controls at the top of the screen, make popup menus come up from the bottom of the screen. This way players can still quickly interact with the UI, but they don't need to block the visuals to do so. This also has the advantage of being more natural for the player as it is much easier to touch and interact with the bottom parts of the screen than it is to reach up and touch the top.

It can be slightly more problematic to redefine how the player interacts with the actual game elements. Most games require the user to block parts of the screen to interact with elements near the top, so developers should consider how much of the screen will be blocked by that action, and how long it will be blocked for. Quick touches or swipes will likely not be a

concern, but if a game requires precise or prolonged drags, then the game should not also require the player to see or respond to something on the blocked portion of the screen at the same time.

When a player is asked to drag or directly manipulate game elements, their finger will likely be covering what they are being asked to control. Unlike a traditional cursor, which takes up just a few pixels on screen, a player's finger will block out a much larger area that completely surrounds the point reported as the touch location. A general rule of thumb is that the area being covered by a finger is a circle with a diameter of around 40–45 pixels.

For most buttons and game UI elements,

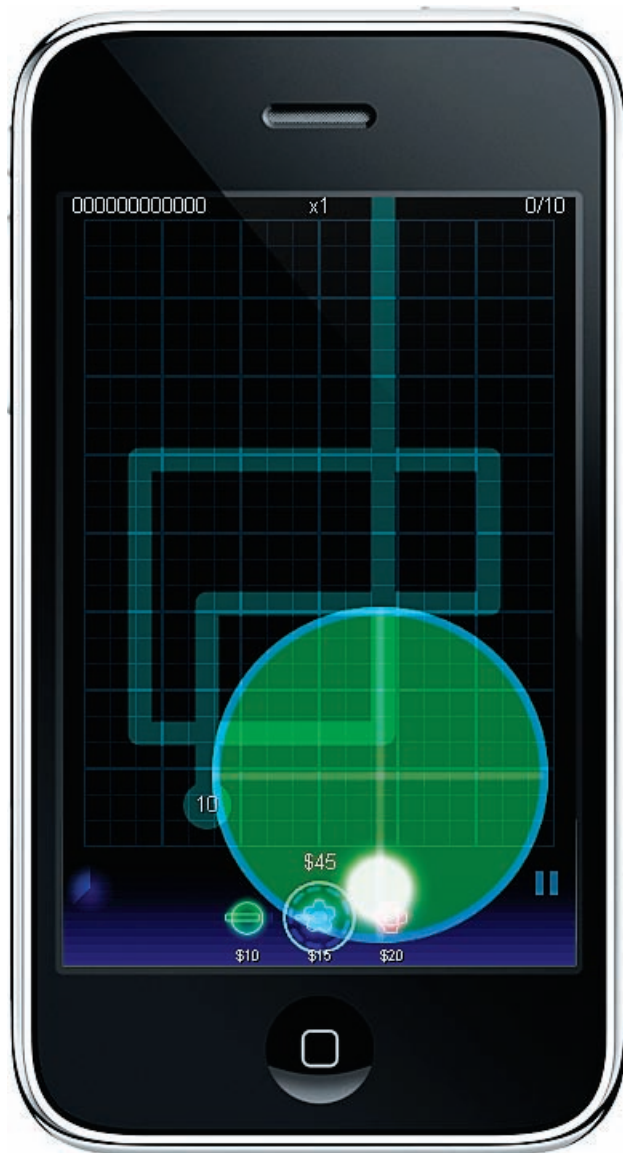
this is not a large concern as the player simply won't be able to see part of the rollover or hit state for the button. But dragging and dropping, or precise placement of game objects by the player, will require a way to deal with the large screen space being covered by the player's finger.

Apple resolves this issue when selecting text by opening a magnifying glass above the user's finger showing what is actually being selected. This magnifying method may work with some game mechanics, but in most cases will likely prove impractical. The developers of SCRABBLE decided not to worry about the blocking of tiles when being placed, which allows direct connection to the piece, and any mistakes are easily resolved due to the forgiving nature of the gameplay.

Another solution that has been used to some success in games like GEODEFENSE is to offset the position of anything being dragged to a position above the touch location, and sometimes to the left or right as well. This allows the user to easily see the object being dragged, and its general location on screen. The trade off is that the user now has slightly indirect control of the object because of the offset, making it a good idea to also show a highlight or similar effect indicating exactly where an element will be dropped. This placement highlight is important to ensuring that users don't get frustrated when trying to manipulate game objects, particularly in grid-based games.

## ALL SHOOK UP

» While touch input causes very obvious visibility issues, less obvious (but just as problematic) are the ramifications of accelerometer control. It



GEODEFENSE.

is very tempting to use large input gestures like those used in many Wii titles to control gameplay. The problem with this is that unlike the Wii, mobile devices have the accelerometer attached to the screen. Thus making large swinging gestures with the device result in the player not being able to see the screen while doing the movement. Some iPhone titles like FLICK FISHING have managed to deal with this quite successfully. However, in most cases it is a recipe for failure. If the game truly does require large movements of the accelerometer, then there are a few things that can be done to help refine the overall playing experience.

First, it is important to make sure that very little is happening on the screen while the player is performing the input gesture. Since the player won't be able to see the screen at that point, the game should pause or slow down the action so that there is nothing the player actively needs to touch or look at while they are performing the gesture input.

Another tactic that can help this immensely is using audio cues to inform the player how they are performing. ISAMURAI uses this to great effect by having sword swinging and hitting sounds play when the desired movement is made, resulting in a game that is entirely gesture-based,

# UI IN THE LAND OF THE TOUCH SCREEN

and which does not require the user to see the screen at all.

While wild accelerometer movement is generally not a good idea for handheld devices, even games that use subtle accelerometer movements will need to find ways to mitigate the movement of the screen while manipulating the accelerometer. Several techniques have emerged to help minimize the impact of screen movement, though no single perfect solution exists today.

Driving games or anything where the player needs to tilt left and right, have perhaps the easiest and best solution to the problem. Whenever the device is tilted to one side, the game camera can be tilted the same amount in the other direction to compensate. This has the effect of keeping the ground level during play, and treating the screen like a window into the game world that the player rotates during play. This has become such a commonly-accepted practice in most driving games, that its absence will frequently be commented on by players and reviewers.

Another suggestion, particularly for games that require movement in all directions, is to not force large movements, and instead allow for smaller tilt motions to have the same in-game effect. The challenge with this approach is that players may find the controls too sensitive and be unable to properly play the game. There is also the issue of making it clear to the player how much movement they need to do, and training them that they do not need to overdo the movements. Many players will assume they need large movements to affect accelerometer based-elements, even when they don't. Accordingly, without proper player education, players may feel that accelerometer-based controls force them to not be able to see the screen, even if that is not true.

## IT JUST DOESN'T FEEL RIGHT

» By its very nature touch input is far less accurate than mouse input. With traditional cursor and mouse-based input, the user can easily get near pixel perfect accuracy, and games can safely require that of their users. With touch input the actual touch location given is the result of a certain amount of guesswork and interpretation logic performed by the underlying systems, before giving the interpreted point to the program. The impact this has on games is twofold.

First, for anything that requires the user to touch a specific location, the hit area needs to be very big, and the logic for detecting hits needs to be very forgiving. Buttons, for example, should have a hit area of at least 40x40 pixels, and game elements should be as close to that size as possible (or larger). For some styles of games, it may be acceptable to go smaller than this, but when doing so developers need to ensure they do enough testing to know that players do not have difficulty interacting with the game elements. One solution to this is to allow users to zoom into the gameboard, so they can reach a point where the interactive



(TOP) FIELDRUNNERS. (MIDDLE) HARBOR MASTER. (BOTTOM) deBLOB calibration progression.

elements are large enough to touch. FIELDRUNNERS handles this exceptionally well through a combination of advanced touch detection logic and allowing users to zoom far into the gameboard as well. For slower paced games this can work quite well, but forcing players to constantly zoom in and out to play the game will become tedious.

For games that require precise placement of game elements, it is also important to compensate for movement that may happen at the end of a touch. As a finger is picked up off of the screen, it is normal for the interpreted touch location to shift by several pixels in any direction. This becomes particularly problematic in grid-



FASTLANE  
STREET RACING

based games, where the user may be near the boundary of one grid location, but as they pick up their finger, the shift may result in the last touch location being in a different grid than they intended. Logic to detect and counteract this movement will dramatically improve this problem and will make users much less frustrated with the results.

For games that use accelerometer input, calibration is extremely important. Each player will naturally want to hold the device at a particular angle, and if the game does not treat that orientation as a neutral starting position, then players may quickly become frustrated at being forced to learn a different way to hold and manipulate their device. Further, each player's natural device orientation may change between plays, making the timing of this calibration step critical.

Ideally calibration should happen as the action is starting, and again any time the player is returning from having the action paused. The games that do the best job of this will typically have a button and possibly onscreen text informing the player what is happening, so they are sure to hold the device at their preferred angle. Not doing this may cause lots of unnecessary frustration. Calibrating too early is especially problematic as many users, particularly reviewers, may set their device down or hold it at an odd angle during load times, and will only hold it properly when the gameplay begins. The calibration screen in DEBLOBS is somewhat involved, but very clearly shows the user what they are doing and gives an indication of how far they will need to tilt to play.

## IS THIS TROUBLE REALLY WORTH IT?

» The best part about developing games for new devices and interfaces is discovering new ways for players to interact with games, which are often as fun as the game itself. It turns out that directly touching and dragging on a screen is an extremely natural way for even casual players to interact with a device, and this innate familiarity results in highly enjoyable gameplay. This is evident through the wild success of games like FLIGHT CONTROL and HARBOR MASTER, which would be incredibly tedious, if not nearly impossible, to play with traditional mouse or joystick input controls.

Similarly, games that require very frequent input on different parts of the screen are also much easier to do with touch and especially multi-touch input, than they would be with traditional controls. Some game types like CASTLE DEFENSE, which would be nearly impossible to play with a mouse, end up being quite manageable with touch input.

While accelerometers may prove too problematic for game control in most titles, if a game relies solely on touch input, subtle accelerometer-based effects can add a lot of perceived polish and depth to the final presentation. Effects like gently moving the camera, or shifting the background when the device is tilted are very simple to implement, but make a big impact. One of the best examples of this is from WORDFU, where tilting the device around will tilt the camera allowing the player to see the letters on the sides of each die on screen.

## CONCLUSIONS

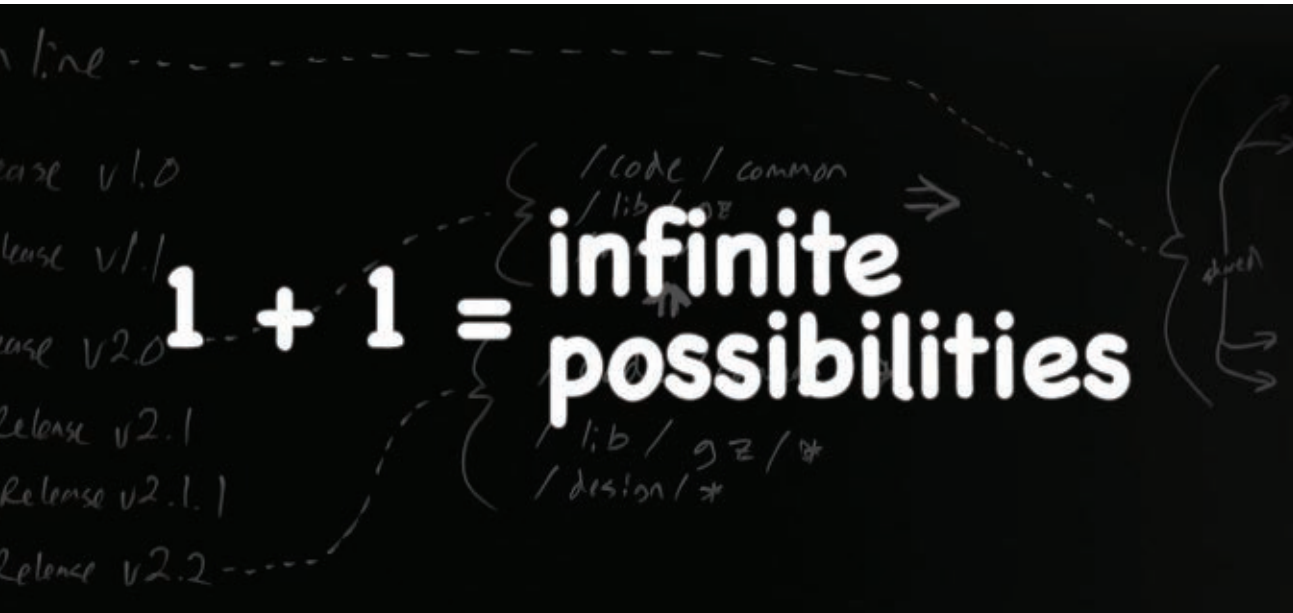
» Today's devices are providing new and better ways for players to interact with games. The degree to which players can viscerally connect to the experience is greatly increased when given the ability to more directly control events. As game developers, it is our responsibility to make sure we harness the new tools at our disposal to enhance the player's experience.

When developing games with these input mechanisms it is imperative to not only consider the big picture elements like how a player's hands will block the screen, but also focus on the subtle results of the player's interaction. Doing this well will provide a much better connection to and enjoyment of the title. 🎮

**BRIAN ROBBINS** is founder of Riptide Games, an indie game developer in Denver, CO. Since 1999 he has programmed over 100 games, and is an active participant in the game development community, currently serving on the IGDA Board of Directors. He can be reached at [brian@riptidegames.com](mailto:brian@riptidegames.com).

[www.seapine.com/gamescm](http://www.seapine.com/gamescm)

Satisfy your quality obsession.



© 2009 Seapine Software, Inc. All rights reserved.

## Get More Change Management with Seapine Integrated SCCM

**TestTrack Pro + Surround SCM = infinite SCCM possibilities.** Seapine's integrated software change and configuration management (SCCM) tools do much more than competing tools, and at a much lower price point. Start with TestTrack Pro for change management and add Surround SCM for configuration management—two award-winning tools that together give you the best integrated SCCM solution on the market.

- Link issues, change requests, and other work items with source code changes.
- Manage simple or complex change processes with flexible branching and labeling.
- Coordinate distributed development with RSS feeds, email conversation tracking, caching proxy servers, change notifications, 3-way diff/merge, and other collaboration features.
- Enforce and automate processes with incredibly flexible work item and file-level workflows.

Built on industry-standard RDBMSs, Seapine's SCCM tools are more scalable, give you more workflow options, and provide more security and traceability than competing solutions.

Get more, do more with Seapine tools. Visit [www.seapine.com/gamescm](http://www.seapine.com/gamescm).

 Seapine Software™

**QA Wizard® Pro**  
Automated Testing



**Seapine CM®**  
Change Management



**Surround SCM®**  
Configuration Management



**TestTrack® Studio**  
Test Planning & Tracking



**TestTrack® ICM**  
Test Case Management



**TestTrack® Pro**  
Issue Management



# AION

## POSTMORTEM





**D**eveloping an MMORPG is a task fraught with peril—doubly so when you're making something that must appeal to both Eastern and Western MMO audiences.

These are demanding users who already have concrete ideas on what they want their next MMO to be. AION sought to bridge the gap with art, gameplay, and story, delivering an MMO experience that would resonate with gamers from Seoul to Soho to SoCal.

Like all MMOs, AION is a work in progress. While we're buoyed by our first year in Asia and our good start in North America and Europe, we're keenly aware of the effort it took to give AION its wings around the world.

From the start, we wanted Aion to be a flying game, with movement on all axes. But we had no idea how easy 2D battles were until we started working with the Z-axis. The flight system consumed double the usual development resources, and even now is somewhat difficult for some players to control. A player's field of vision is necessarily limited on the ground, but up in the air, the number of visible game elements expands exponentially. Some players have wound up using the flight system for exploits, but we've felt the results were more positive than negative. Indeed, like Icarus, we flew ...

#### WHAT WENT RIGHT

**1} CRYTECH'S CRYENGINE—FLEXIBLE ENOUGH FOR AN MMORPG.** We used an engine optimized for first person shooters in an MMO because we really wanted AION's environment to "pop" off the screen. CryEngine handles landscape textures with ease, and it's great at processing light effects, which is particularly important in a world where available light tells part of the game's story (and AION is based on the world of angels).

That said, CryEngine isn't an MMO engine per se. We had to rearrange about 90 percent of the engine to make it work for AION. The building blocks were there; they just needed to be reassembled. The team has since worked hard to improve the graphical quality without switching graphics engines. Whereas before our artists might have simply designed something to be as gorgeous as possible, as we become even more familiar with the technology we are finding ways to improve quality without any major performance or polygon hits.

**2} ART AND ANIMATION.** Western and Eastern gamers have sophisticated tastes, and they don't always match up. Art-wise, Western gamers prefer heavier lines and rougher textures, while Eastern gamers prefer cleaner lines and characters that border on "cute" or "cartoon-like." We wanted to satisfy as many players as possible, so that meant walking a tightrope in terms of art direction, but allowing some specific detours into more Eastern or more Western styles.

This meant that our character customization system necessarily grew to an enormous size to compensate for all tastes—but the players

# AION

found it to be enormously fun. MMO players spend countless hours staring at their characters, whether they're fighting, exploring, or just using in-game chat to talk to their friends. We really wanted to make even "idle time" evocative of the world. AION characters shiver and breathe steam when it's cold outside, and they wipe their brows when it's hot. If they're left idle in shallow water, they try to grab the fish swimming past. When it rains, they get out umbrellas. Those animations paid big dividends; they're an effective way to demonstrate that ever-elusive "polish" in an MMORPG.

**3} THE BAN BUNNY.** All MMOs experience some form of trouble with real money trading (RMT), and AION is no exception. For those who don't operate MMOs, this is when players will farm for in-game gold exclusively, and then sell this gold to players for real-world cash, which is

illegal in many countries, and certainly does not help create a fun game environment. Real money traders are difficult to identify, and even when you do, it's hard to know what you should do about them. AION took a unique approach.

We have a team called the Game Surveillance Unit, or GSU. The GSU's job is to identify and eliminate these goldfarming RMTs, bots, and chat spammers. The big difference between what we do, and what a surveillance unit like FINAL FANTASY XI does, is we eliminate them publicly. Rather than quietly deleting them from our servers, our GSU spawns in a giant rabbit, fire breathing squirrel, or other odd creature to literally blow the offending bot or character out of the world. This not only shows the RMTs we mean business, it also shows the players

we care about this issue, as they're all able to witness the RMT's destruction.

#### 4} A BIG, BOLD SOUNDTRACK.

Another demonstration of polish was the largest-scale original soundtrack in the history of Korean games. Composer Yang Bang Ean spent three years directing the soundtrack. He comes from an interesting background, as a Japan-born Korean who originally studied as a doctor, and in fact worked for several years in a Tokyo hospital [in Japan he is often credited as Ryo Kunihiko].

He began learning piano at age 6 at the Tokyo National University of Fine Art and Music, has since released several albums, and scored animation, films, and games. He was perfect for our multicultural approach, as one of his parents comes from North Korea, and

the other from South Korea, and he was born in Japan, while being of Korean descent. These origins inspire his culture-bridging music, which was important to our East-meets-West product.

AION's soundtrack was recorded with the London Symphony Orchestra, and engineered at the famed Abbey Road Studio in London. A soundtrack CD was included in the AION SPECIAL EDITION package, emphasizing the importance of unique, iconic music to the overall AION experience.

Additionally, there are a total of 6,000 types of sound effects in AION, including 650 musical pieces, of which 50 were played during battle.

#### 5} EASY TO LEARN, DIFFICULT TO MASTER.

This was a bedrock principle from the very start—and it survived unscathed through all the development-team turnover in AION's early months [more on that later]. We wanted AION to be

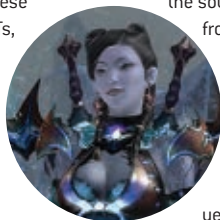
approachable for gamers who had never tried an MMO before, so we intentionally moved both key MMO tropes [crafting, skill/power decisions, PvP, group instances] and AION-specific elements [branching chain attacks, flight, access to the Abyss] later in the game.

Players don't fly until they reach level 10, for example, and they can't access the Abyss [where flight is largely unlimited] until level 25. Veteran MMO players will get their wings in a matter of hours and will be swept up in PvP battles in a matter of days. But AION takes great effort to manage the flow of complexity, opening the valve one gameplay element at a time.

#### WHAT WENT WRONG

##### 1} DEVELOPMENT PART SOAP OPERA, PART MEATGRINDER.

We knew early on what we wanted AION to be—a global MMO that could reach across the Pacific. But how to get there from here? Those first





tentative steps—the basics of the world and gameplay—were steps we took over and over again. We'd settle on a design direction, then start questioning it almost immediately. Every time we faltered after a few steps, morale plummeted.

In late 2007 we announced that the AION closed beta test (CBT) was going to be postponed for three months. Because NCsoft was banking on AION to breathe new life and excitement in the company, the morale of the entire organization was at risk, and the health of the development team was compromised. To add insult to injury, the biggest consequence at that time was the fact that stock price fell because of the postponed CBT.

It wasn't just a question of developers feeling disillusioned with their work. The development team was replaced three times in four years. The game got a complete reboot in 2006, and at that point the game was mandated to

complete very soon thereafter. And even surviving that churn wasn't enough. Two producers in charge of AION quit due to stress-related health issues. The pressure was intense, and AION was eating us alive.

## 2} COMPARISONS WERE INEVITABLE, BUT EARLY COMPARISONS WERE TOXIC.

The pressure on the developers wasn't just internal. Almost as soon as it was announced, gamer media gave AION the "WoW-killer" tag. That inheritance was unwanted but probably inevitable, because there hadn't been a truly global MMORPG success since WORLD OF WARCRAFT. Comparisons with both Western games (HELLGATE: LONDON, WARHAMMER ONLINE) and Eastern games (PRIUS ONLINE) soon followed. Even before launch—even before we knew what we were going to launch—we were in competition with a half-dozen great games.

We make games, so we know firsthand how competition

can be healthy; it's a virtue in and of itself. But those comparisons came so early that it was hard to ignore the other games and figure out what we wanted AION to become. There may be a place for reactive game design, but it's not at the beginning of the development process.

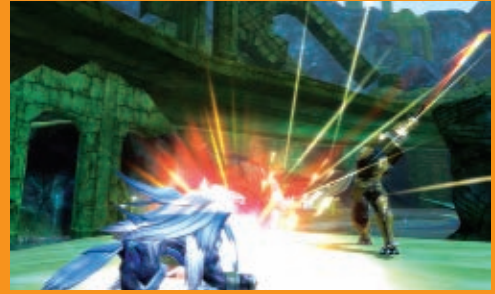
## 3} IS THAT YOUR VOICE?

It was early 2008 before the CBT took place. We had not yet integrated voice into the game, so our developers were taking turns recording the voices. Almost all the voices (including PCs, Shugos, children, the elderly, and even villains) were recorded by the current team leader in charge of AION, and eleven of the female voices were recorded by a colleague in the development team (luckily a female) to fill in the missing voice actors.

The problem popped up abruptly when we needed a key voice recording. We had a demo test the following day, and needed the voice of "Ariel," one of the goddesses

## aion users by gender group

(as of august 2009, South Korea only)



age	male	female	total
10+	6.1%	1.1%	7.2%
early 20s	17.0%	4.0%	21.0%
late 20s	20.4%	5.0%	25.4%
early 30s	16.0%	3.0%	19.0%
late 30s	9.0%	2.0%	11.0%
40+	6.0%	3.0%	9.0%
50+	2.4%	3.0%	5.4%
60+	1.0%	1.0%	2.0%
total	77.9%	22.1%	100.0%



# AION

## aionfacts



**133**

CURRENT NUMBER OF SERVERS FOR AION (WORLDWIDE)

**60**

NATIONS IN WHICH THE GAME IS DISTRIBUTED

**7\***

NUMBER OF LANGUAGES SERVICED

\*Korean, English, Japanese, French, German, Chinese, and Russian

**43,269**

AVERAGE NUMBER OF CHARACTERS CREATED DAILY DURING AION'S FIRST YEAR

**3:2**

RATIO OF MALE TO FEMALE CHARACTERS CREATED

**2,500,000**

TOTAL NUMBER OF WORDS IN AION

**31,000**

NUMBER OF ITEMS IN AION

**16m/s**

MAX FLYING SPEED

**50 meters**

FATAL FALL HEIGHT

**228,336**

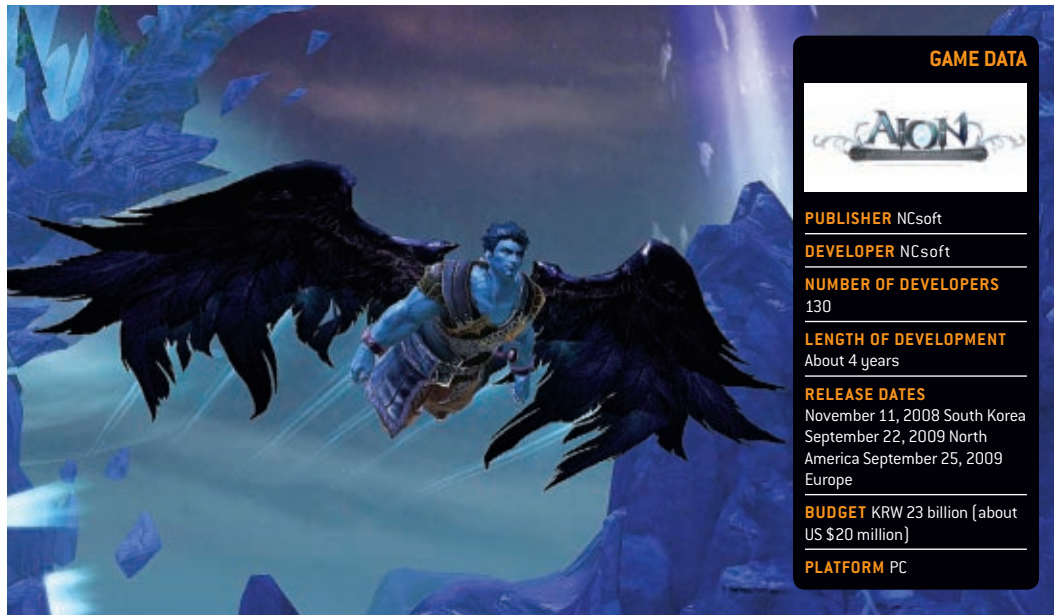
NUMBER OF LEGIONS (AION'S GUILDS) CREATED SINCE THE OBT

**10:1**

RATIO OF AION'S LAND MASS TO SEOUL'S

**2 real-world hours**

LENGTH OF ONE FULL 24 HOUR CYCLE IN AION'S WORLD



## GAME DATA



**PUBLISHER** NCsoft

**DEVELOPER** NCsoft

**NUMBER OF DEVELOPERS**  
130

**LENGTH OF DEVELOPMENT**  
About 4 years

### RELEASE DATES

November 11, 2008 South Korea  
September 22, 2009 North America  
September 25, 2009 Europe

**BUDGET** KRW 23 billion (about US \$20 million)

**PLATFORM** PC

in AION. The female developer who was going to record the voice was so ill she caught laryngitis and could not make any sound at all much less record a voice over.

Time was running out so we tested with other female developers, but because the recording had more to do with the "acting" rather than the "voice" itself we were all becoming desperate. The best idea we could come up with at the time was to record the team leader's voice and change it digitally. As soon as this idea came up we put it in action. Of course, the team leader recorded it by himself (if somebody else saw or heard it at the scene, it would have required a lot of explaining).

Once recorded in the male voice we changed it to female sound range (about one octave difference), then added additional plug ins and programs to control the timbre, tone and speed of the monologue.

After the demonstration, people started asking the team leader if this was actually his voice. Moreover, some gamers after the CBT left very negative replies regarding the dubbing of voices. What they don't know is that even now, one

year after the South Korean launch, there are still lots of voices recorded by the same team leader. In the end, 30 professional dubbing artists left their voices to the NPCs.

### 4} IT'S HARD TO SERVE TWO MASTERS.

One of our early design goals was to make a game that would support a "pyramid" of users: a wide base of more casual players, narrowing upward to a minority of hardcore users that would play AION with real fervor. That's easier said than done, of course. What's "boring" to the hardcore player is "impossibly frustrating" to the casual gamer. This problem was compounded by the high percentage of new or young developers on the team, who had less experience dealing with these sorts of design issues.

In the end, we made a conscious effort to stay away from the large-scale endgame raiding that's traditional in MMOs, opting instead for faction-based warfare in the PvPvE structure of a single mega-zone called the Abyss to keep the hardcore happy and the casual players engaged. Zone-based combat allowed us to concentrate some of these experiences in certain

areas, which also helped manage the userbase.

### 5} WE FELL INTO THE ABYSS.

Once we settled on the Abyss as our way of building that pyramid structure, our problems were only beginning. Once again, we'd come up with a good answer that proved maddeningly difficult to implement.

The Abyss is triple the size of AION's other zones, and it's far more active. It's a realm of unfettered flight and omnipresent PvP. Fortresses and Artifact sites scattered all over the zone deliver faction-wide buffs when controlled. And the zone has a third "faction"—mobs (AI-controlled monster opponents) that provide PvE challenges throughout the zone and bring the Abyss back to equilibrium if one faction dominates.

One of the biggest challenges for us was the reduction of the computer requirement. It proved to be extremely difficult and we weren't even sure we could accomplish the task. One of the characteristics of an MMORPG is that you have massive amounts of people concentrated in random parts of the maps. These situations take up extreme resources and also

produce a large amount of lag in the game. We put a lot of effort toward maximizing both the beauty of the game and smooth gameplay when large amounts of people gathered in one place, particularly during the large scale battles in the Abyss.

That's a lot of moving pieces, and ensuring they were well-behaved pieces was an arduous forced march.

### FLIGHT OF FANCY

» AION's been out for more than a year in South Korea, but just a few months in North America and Europe. Yet in all regions, it feels like our flight has just begun. There's nothing like a project with a big, bold vision: Make a truly global MMO—something that crosses all oceans and delivers a satisfying experience no matter what your "gamer heritage" is.

Just getting the game out the door is victory enough. But with an MMO, you don't have a release so much as you have a launch. We've got big plans for what's next for AION, but we're also learning a lot from our players about what they want AION's future to be.

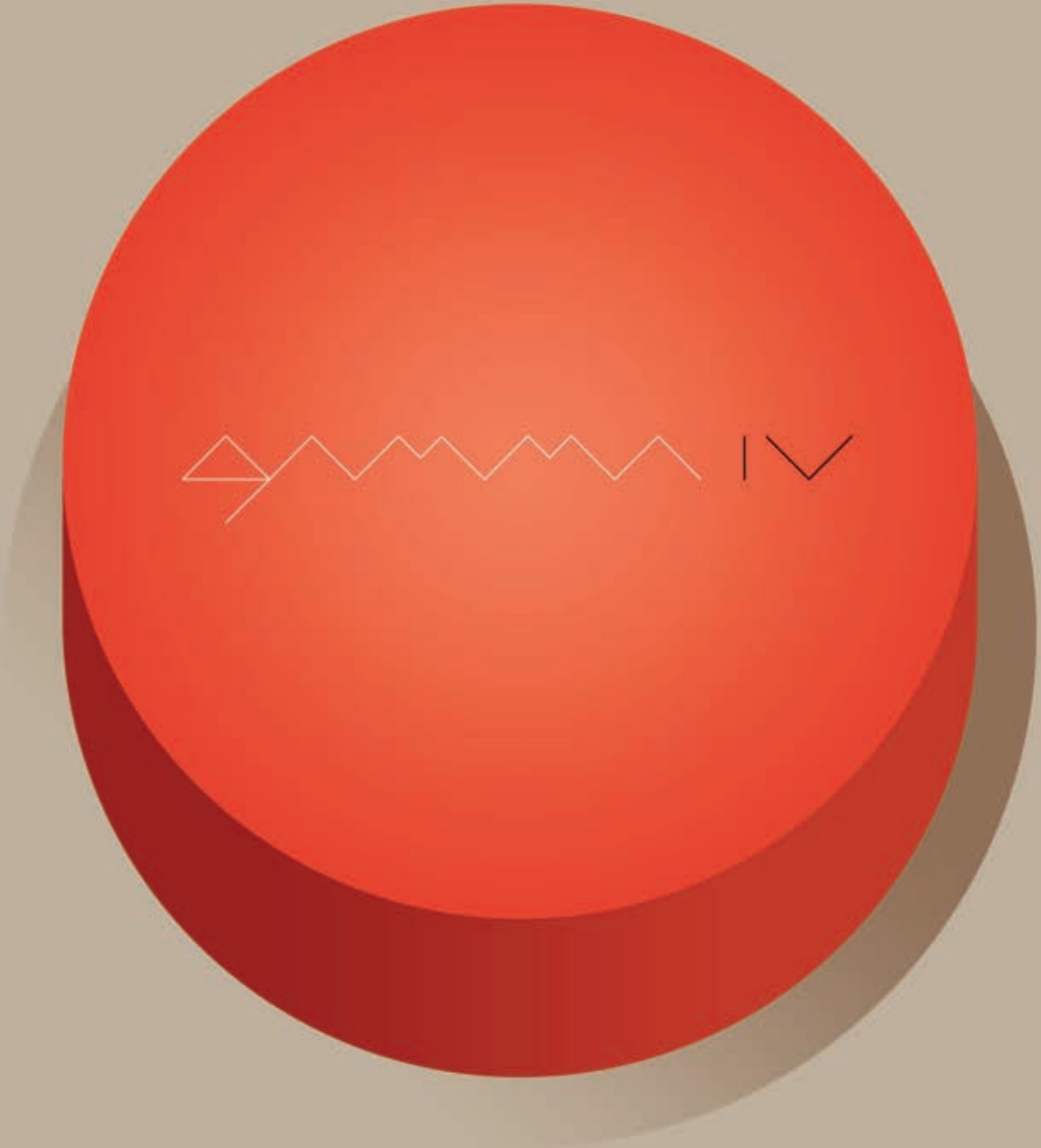
### NCsoft DEVELOPMENT

TEAM Seoul, South Korea.



GAMMA IV  
 ONE BUTTON GAMES PARTY  
 THE MEZZANINE  
 444 JESSIE STREET  
 SAN FRANCISCO  
 MARCH 10 2010

GAMMA IV  
 ONE BUTTON GAMES SHOWCASE  
 GDC EXPO  
 MOSCONE SOUTH  
 SAN FRANCISCO  
 MARCH 11-13 2010



GAMMA IV PARTY MADE POSSIBLE BY:

[WWW.KOKOROMI.ORG/GAMMA4/](http://WWW.KOKOROMI.ORG/GAMMA4/)





# Unreal Technology News

by Mark Rein, Epic Games, Inc.

Canadian-born Mark Rein is vice president and co-founder of Epic Games based in Cary, North Carolina.

Epic's Unreal Engine 3 won Game Developer magazine's Best Engine Front Line Award for three consecutive years, and it is also the current Hall of Fame inductee.

Epic's internally developed titles include the 2006 Game of the Year "Gears of War" for Xbox 360 and PC; "Unreal Tournament 3" for PC, PlayStation 3 and Xbox 360; and "Gears of War 2" for Xbox 360.

## Upcoming Epic Attended Events:

### CES

Las Vegas, NV  
January 7-10, 2010

### DICE Summit

Las Vegas, NV  
February 17-19, 2010

### GDC 2010

San Francisco, CA  
March 9-13, 2010

### Triangle Game Conference

Raleigh, NC  
April 7-8, 2010

Please email:  
mrein@epicgames.com  
for appointments.



## OVER 170 FREE VIDEO TUTORIALS FOR UNREAL DEVELOPMENT KIT AVAILABLE NOW

Epic Games recently launched the Unreal Development Kit (UDK), a free edition of Unreal Engine 3 that provides uncompromised access to the award-winning toolset.

In addition to publishing hundreds of pages of technical documentation for UDK, Epic has released upwards of 170 new video training tutorials for free download at the Unreal Developer Network ([udn.epicgames.com](http://udn.epicgames.com)).

Those interested in using UDK for commercial purposes should refer to [www.udk.com/licensing](http://www.udk.com/licensing), which has been updated to answer many commonly asked questions.

## 3D INTERACTIVE BUILDS CORPORATE AND MILITARY SIMULATIONS WITH UNREAL ENGINE 3

3D Interactive licenses Unreal Engine 3 to develop high-fidelity simulations for global construction and mining companies - most notably, Caterpillar - as well as the U.S. military.

As 3D Interactive Technical Director Andrew Czarnietzki points out, "A simulator allows training to happen safely, without burning real fuel or taking an expensive machine off the production line. A student who has spent eight hours on a simulator before first touching a machine has a significant advantage compared to a student who has only sat through a classroom lecture."

"Unreal Engine 3 allows us to build simulators with a high level of visual, physical and functional realism while keeping final unit costs low," says Czarnietzki.

"This high-quality, low unit cost model allows us to provide simulation-based training towards machines that would not normally warrant the multimillion dollar, flight simulator-style units. A low unit cost also allows us to alleviate training bottlenecks."

Czarnietzki explains how Unreal Engine 3 empowers his team to focus on the overarching design of its training simulators and avoid getting hung up on subtle details.

"Unreal Engine 3 provides the systems and tools that enable us to achieve an incredible level of authenticity while focusing development efforts on the simulators themselves. Small intangible details, such as heat distortion in the exhaust or rope ladders that swing with physics, can be easily enabled."

"I absolutely love the material editor. The what-you-see-is-what-you-get, node-based system is an absolute dream. I've been able to get some really incredible results, though perhaps not in the way Epic had ever designed, and these results are central to our visual realism," says Czarnietzki.

"In addition to the material editor, we use the skeletal control system as a base to our rather complicated vehicle simulation technology; with the two working together, we can quickly connect and rig complex linkages, flexible hoses, hydraulic systems and all the other core components to heavy equipment. We recently started an oil field project, and I was personally blown away at how fast all the complicated hydraulics and pistons all came to life."



3D Interactive's 924H simulator built for the U.S. Army

While its projects are large in scope and require tight timelines, 3D Interactive's core team of developers has mastered how to efficiently work with the engine to maximize productivity.

"We have four programmers, four artists, a technical director who bridges the gap and a handful of part-time contractors," says Czarnietzki.

"Compared to most game projects, our development time is extremely short. For example, we built and delivered the entire U.S. Army's Caterpillar Small Wheel Loader simulator in under five months. Without UE3, I would expect our development times to have doubled and the quality of our simulators to have been compromised. The solid foundation provided by Unreal is central in our ability to building and delivering high-quality simulators quickly and efficiently."

"When we were considering various engines, the biggest question was, 'Do we really need Unreal?' Since we have started using it, though, I think the only answer is that it is worth every penny."



For UE3 licensing inquiries email:  
[licensing@epicgames.com](mailto:licensing@epicgames.com)

For Epic job information visit:  
[www.epicgames.com/epic\\_jobs.html](http://www.epicgames.com/epic_jobs.html)

WWW.EPICGAMES.COM



---

# CENTER OF MASS

## TOOLS AND TECHNIQUES FOR ANIMATING NATURAL HUMAN MOVEMENT

---

**A COMMON COMPLAINT FROM CONSUMERS** regarding computer animation (especially in the video game industry) is the “puppet-like” movements characters often seem to perform. When two character models come into contact with each other in a fighting game, or when they must climb structures in an acrobatic manner, the difficulties become more pronounced. In this article, I'd like to talk about some techniques for analyzing human movement that can be applied to make your hand-keyed character animations look more natural.

### TAKES ONE TO KNOW ONE

» Before joining the 3D computer software industry I was a competitive gymnast, and also worked as a gymnastics coach. When I was a gymnast, I would practice a move over and over to achieve the right timing, speed, balance, and angle. I repeated and adjusted my technique until I could execute the move perfectly every time. I developed my skill to perform, analyze, and teach gymnastic moves through those experiences.

While I was coaching, I also studied physiology and kinesiology for sports science in university, focusing on the mechanism of the human body in movement. I learned how to analyze the physics of human motion using mathematical calculations, which lead to an understanding of how muscle generates the force to perform a move.

I was hired by Kung Fu Factory to look over the animation team to help them improve the quality of the company's character animation. Over the years I've come to realize that animating characters in 3D space is similar to coaching gymnastics, and my analysis of movement was quite relevant to the creation of quality animation. If you can analyze a move and understand the kinetics behind it, you can create better, more natural animation—and faster, too.

### THE WHAT AND WHY OF CENTER OF MOVEMENT

» Some skilled animators have the ability to picture and recreate human movement easily. But most of us can't. At our studio we often encourage animators to use reference videos showing a person performing the action they're meant to animate. This helps them understand the movement better, but video can only go so far.

One critical, and often undervalued element of movement is an understanding of the character's center of gravity, or center of mass. I will refer to both as the “COM.”

The center of gravity is the point at which the entire weight of a body may be considered as concentrated, so that if supported at this point the body will remain balanced in any position. Your balance and movements are always affected by gravity. As a gymnast I learned how

# CENTER OF MASS

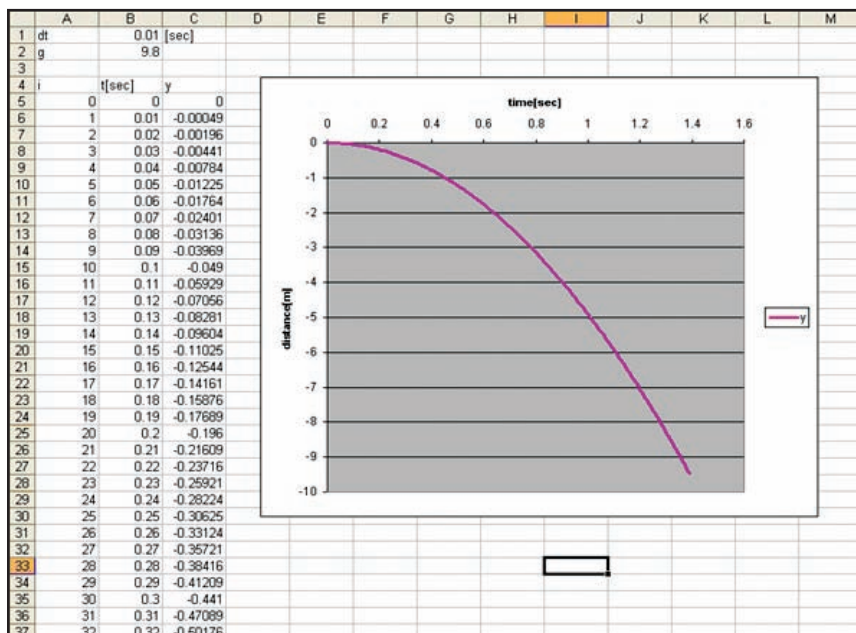


FIGURE 1 Examining free fall.

to control and adjust my body's COM to perform various actions. Tumbling, balancing, and so forth, all require an adjustment of the COM. Some animators I know who are martial artists also understand this concept, so they can see how the COM flows and adjust the character's body appropriately. Adjustment of the COM is something we all do naturally in real life when performing actions like dancing, running, and so forth—but it's not easy to create this in a fictional character.

Figuring out the position of the COM is not hard when a human is standing up straight. Usually the COM of a human body sits around the lower stomach, which is a little above half his height. On a moving character it becomes more difficult, because the location of the COM will constantly shift. For example, as you bend forward from a standing position, your COM will start to adjust and move forward as well. Eventually your COM will be so far forward that you'll fall over if you don't do something to regain your balance. This shifting of the COM position during the move represents the main flow of the movement you're performing [in this case bending over]. We call it the COM trajectory.

If you can properly animate your characters with the appropriate COM trajectory, they will move much more realistically.

To do this, you have to know the physics behind the movement being performed, and how physics relates to the center of mass. An obvious example is gravity. Every object on Earth is affected by gravity. To understand how gravity affects the COM, physics calculations can be very useful. When I work with animators, I use Excel

to show the physics simulation and how the movement is affected by gravity.

In the following section I will show you my process of physics simulation using Excel. I will also show a tool that displays the COM trajectory, used to examine if the COM is moving naturally.

## PHYSICS SIMULATION USING EXCEL

» When you animate in 3D, you will manipulate three transform controllers (translation, rotation, and scale). Each of these controllers can be broken down into x, y, and z components. Understanding the relationship between these components is very important.

In games, these movements may need to be exaggerated—but understanding physics gives you the foundation to create a realistic base. From this base, you can develop or adjust to your animation to meet your expectations for the movement performed.

The advantage of using Excel is that once the formula is set up properly, the calculation result is updated immediately when variables are updated. Excel can also export a visual representation of how the COM should move in the form of a graph. By doing this, COM translation and rotation is broken down into x, y, and z components, which can be used as a visual aid for reference.

The components will be described on the x, y plane (we'll ignore depth) as x = horizontal and y = vertical.

## FREE FALL: GRAVITY AND ACCELERATION

» Characters in 3D software don't have weight. To make something look like it has weight, you'll

need to simulate gravitational acceleration. The easiest example is a free fall.

When an object is falling with no acceleration other than what's provided by gravity, the falling distance (y) could be calculated with this formula:

$$y = -\frac{1}{2}gt^2$$

g = gravitational acceleration (= 9.8)

t = time (sec)

y = distance (m)

Figure 1 shows this formula in a line chart. Calculating this curve will give animators a good base for creating a realistic fall animation. Take the same acceleration and apply it to the vertical translation channel in the 3D software (which is Z, but sometimes it can be Y depending on the product).

As you see in the formula, this is movement with constant acceleration (g = 9.8). You can see the distance (y) within each time increment is growing little by little. This formula is common knowledge but it's not always utilized for the creation of game animation. If the acceleration is not constant, the fall will look jerky.

The most accurate way to animate proper acceleration is to animate an object frame by frame. If you want to use fewer keys, try adjusting the tangents.

## TOPPLE: INVERTED PENDULUM

» If character A knocks out character B with a strike, character B's body falls to the ground, and gravitational acceleration is applied. If the animation doesn't accelerate when the body is falling, the animation will look "floaty" and will give the impression that the object has no weight.

Calculating the acceleration in this case is not as easy as calculating the gravitational acceleration, because the foot is pinned to the ground. This foot becomes a pivot point and the COM will rotate around the pivot. This results in the COM trajectory being shaped like an arc. This mechanism is complicated, so let's break it down into a diagram. See Figure 2.

This movement could be described as an inverted pendulum with a static pivot. In this case, the distance between the pivot (foot position) and the COM will be defined as the pendulum length (l). For the calculations I'll need the following components.

- 1)  $\theta$  (angular position calculated by acceleration)
- 2) X component (Horizontal translation of the COM)
- 3) Y component (Vertical translation of the COM)

First I will calculate  $\theta$ . This is the differential equation that determines acceleration.

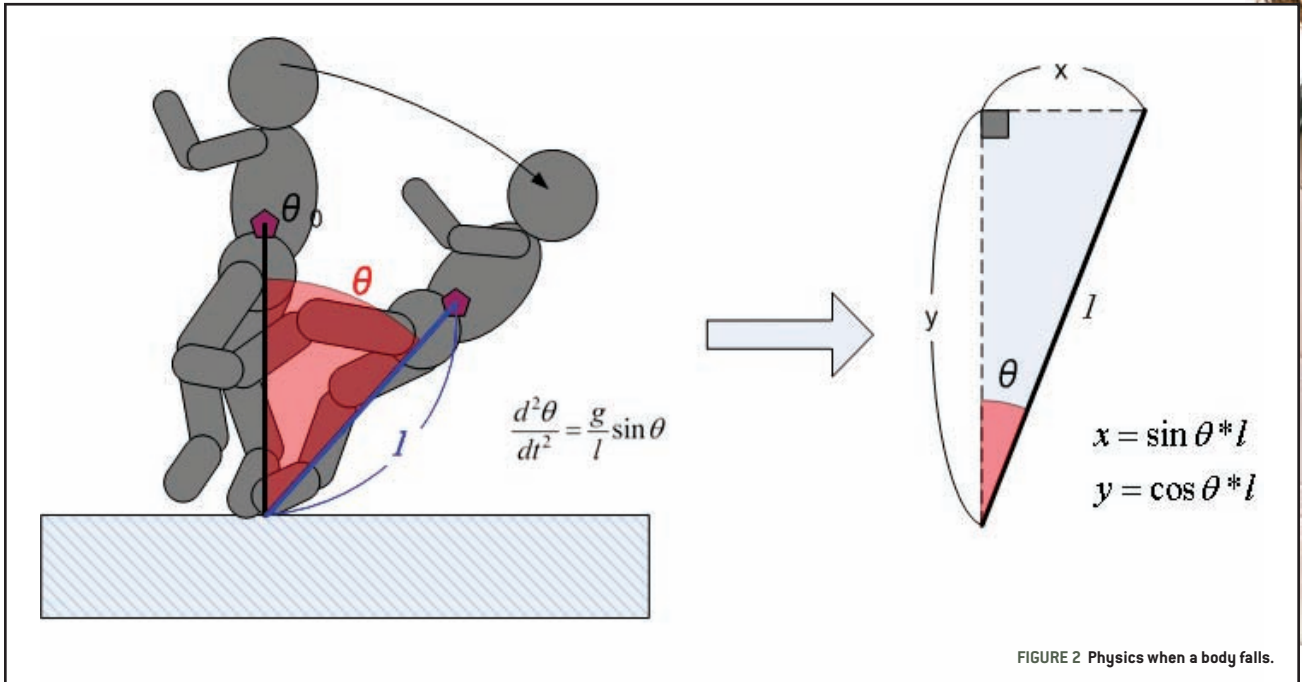


FIGURE 2 Physics when a body falls.

$$\frac{d^2\theta}{dt^2} = \frac{g}{l} \sin\theta$$

---- acceleration

$g = 9.8$  [gravitational acceleration]  
 $l$  = the length of the pendulum  
 $\theta$  = angle [radian]

The first integral of motion found by integrating the formula above is:

$$\frac{d\theta}{dt} = \sqrt{\frac{2g}{l} (\cos\theta_0 - \cos\theta)}$$

----- velocity

$\theta_0$  = Initial angle to start

Integrating velocity will return the angle ( $\theta$ ). The result of the simulation is shown in Figure 3.

Once I get the angle  $\theta$ , the x and y components can be calculated from trigonometric functions, as below. This is also calculated for each time unit in Figure 3 as well.

$$x = \sin\theta * l$$

$$y = \cos\theta * l$$

When length = 0.6, if the fall starts from an initial angle of 5 degrees, the angle changes over time as seen in column C of Figure 3 [green border]. The value is in radian not degrees. The x,

y coordinates are shown in column G (red border) and H (blue border).

The graph on the top right in Figure 3 shows the x, y coordinates. This could be used as a reference to determine the COM trajectory in 3D space. The bottom one shows  $\theta$ . This could be the rotation curve for the root bone.

Note that I used a simple rectangle rule numerical integration in excel to calculate the  $\theta$ . When you calculate a numerical integration in this way, the calculation is most accurate when the time interval is small. See Figure 4 for the comparison between the results with two interval steps.

In Figure 4, the left side of the table shows when the interval step is 0.01—the right side shows the interval at 0.0333. When you compare the time it takes to reach 90 degrees (falling flat to the ground), the time[t] using the interval of 0.01 is 1.320 sec, which is about 0.08 sec faster than the time[t] using the interval of 0.0333. Thus, if you do the simulation with smaller steps, you can get a more accurate result.

If you perform the same calculation in a 3D package, when it's 30fps you would get the same result as the right table, which is not as accurate as the one on the left. If you need to obtain more accurate values, it's ideal to set up "sub-steps" between each frame to calculate a more precise integration.

The same simulation can be done using max script (see Figure 5). The comparison between three different pendulum lengths shows that the falling speed (velocity) is inversely proportional to the length[l].

#### EXAMINING COM TRAJECTORY USING THE COM TRAJECTORY TOOL

» The result of the simulation in Excel tells you how the COM moves, and you can animate your character's COM to match the result. When you do this, you should use the root bone to move the character, not the COM itself because the COM is not an object that you should manipulate, but rather an animation guide.

Some animators that I've worked with were confused because they believed that "root equals COM." But that's not quite right, because the COM is the center of gravity, and it moves around depending on posture. The difference between COM trajectory and root trajectory is obvious when the body is rotating. See Figure 6.

In this figure, a boy is doing a back flip. As you can see, the trajectory of the root bone is completely different from the COM trajectory. Without knowing this fact, if you animate the root to match up with the COM, your movement would be incorrect.

To help with this, I created a COM trajectory tool to calculate the coordinates of the COM of animated characters, which shows the difference between the trajectories. You can also use this tool to evaluate whether the COM trajectory of the character animation is moving naturally.

#### COM TRAJECTORY TOOL SPEC

» When a character is animated, the posture changes from frame to frame which also causes the COM to move constantly. To calculate the COM trajectory, we calculate the mass of the character's mesh frame by frame.

CONTINUED ON PAGE 30

# CENTER OF MASS

CONTINUED FROM PAGE 29

In reality, the human body is structured with a lot of different components like fat, bones, and muscles, and each component has different density (fat is lower, bone is higher). Also, the composition of bodies will vary between individuals.

For our math, we are assuming that the typical human body is roughly the same average density as water (1g/cm<sup>3</sup>). In this way we can consider an object's weight (mass) to be equal to its volume. After calculating that the object's volume (= mass), the COM of the object can be calculated with several formulas.

I will explain the concept using a simpler shape; an octahedron. See Figure 7.

To calculate the octahedron's mass, I am going to divide it into eight tetrahedrons for easy calculation. To do that, I placed an arbitrary point (P) inside the octahedron so that it will form eight tetrahedrons between each face and the arbitrary point. The sum of mass of the tetrahedrons will indicate the mass of the octahedron. The mass of each tetrahedron (Mn) can be calculated using the coordinates of three vertices and an arbitrary point.

Note that in order to get the vertex coordinates that shape the triangle, the order of the vertex has to match with its vertex index so that it can calculate the volume properly.

After the vertices' coordinates are determined, the tetrahedron volume can be calculated with the following formula:

$$Mn = \frac{1}{6} (V2 - P) \times (V3 - P) \bullet (V1 - P)$$

The volume of the octahedron [M] is the sum of the tetrahedrons' volumes. Next, the center of mass [Cn] of each tetrahedron can be calculated with the following formula:

$$Cn = ((V1 - P) + (V2 - P) + (V3 - P)) / 4$$

After Cn is calculated, multiply Cn with Mn for each tetrahedron. Add those values together, then divide it by the weight of the octahedron [M]. This will return the COM of the octahedron [as in the formula below]. This can be extrapolated for more complex objects, like character models.

$$C = \sum_{i=1}^n (Cn_i * Mn_i) / M$$

To get the COM trajectory, this process should be repeated through the length of the animation. For our company, we have a tool that draws a spline to form the COM trajectory. We use this tool to evaluate whether the flow of the COM is translating properly.

This script could be used to examine paired animations as well. See Figure 8, which depicts a hip toss.

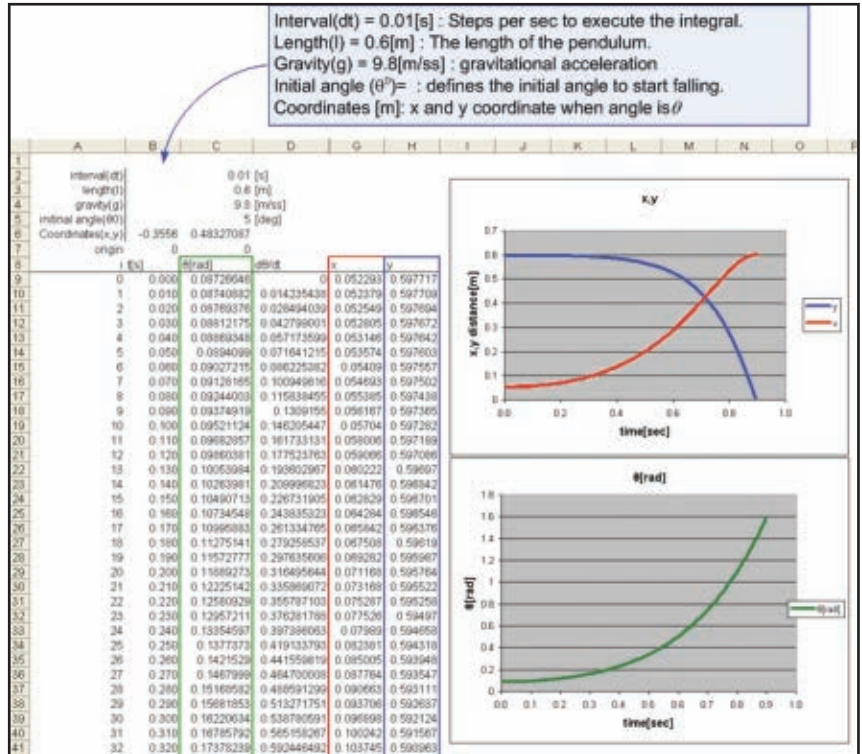


FIGURE 3 Numerical integration of the inverted pendulum.

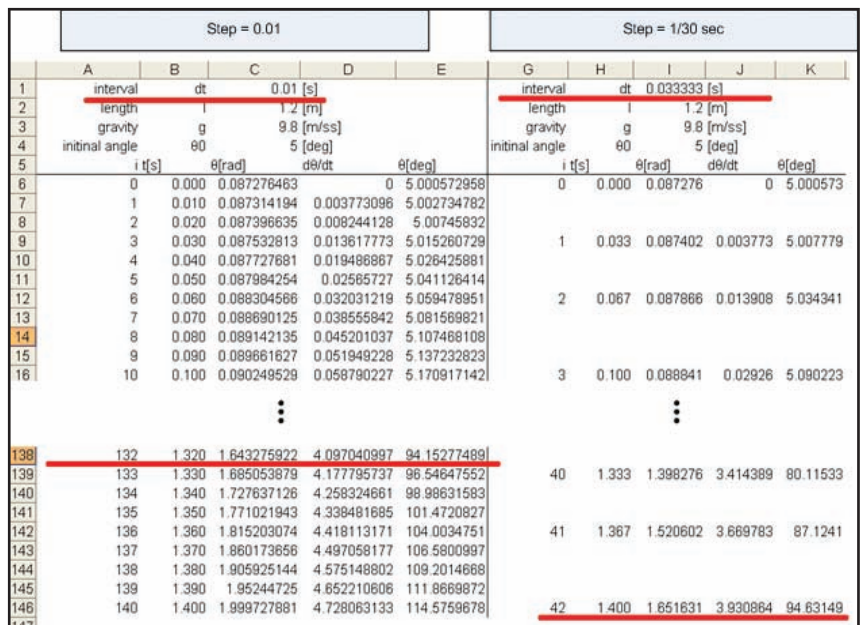


FIGURE 4 Comparison of the integral using different intervals.

The hip toss is a move that's difficult to animate, because two bodies are falling together. The red line is the COM trajectory of the attacker, and the blue line represents the trajectory of the defender. The two bodies are moving together

like one object, so they form a compound object [attacker + defender]. The COM of the compound object can also be calculated by using the last formula above. The green line you see in the figure is the compound COM trajectory.



In the evaluation process, this tool can be used to create a tangible reference so that animators can adjust their animations.

**COM TRAJECTORY TOOL USE**

>> To give an example of how you might apply this tool—one of our animators had created an animation where the character was performing somersaults as he was falling, almost like an Olympic competitive diver. The animation looked off but the animator didn't understand why.

The problem was that he had animated the root as if it were the center of rotation. But the COM position when the human body is in this pose would be closer to the front of abdomen outside of the body. In this case the body should be rotating around the COM, not the root bone, which would cause movement to look wobbly and unrealistic.

Figure 9 shows this scenario. Here, the red lines represent the character's root bone trajectory, and the blue lines represent COM trajectory. By using the tool, you can see the COM moving from side to side in the right picture, which is physically unrealistic. The image on the left shows how gravity pulls the COM straight down while the root is rotating around it.

In a wire action movie, a wire is connected to the stunt person's hip to help him or her perform movements in the air. Because of that, the center of rotation is fixed to the hip position. In that case, the motion would look like the picture on the right in Figure 9, so the rotation "wobbles," and does not smoothly translate. This can happen if you try to simulate

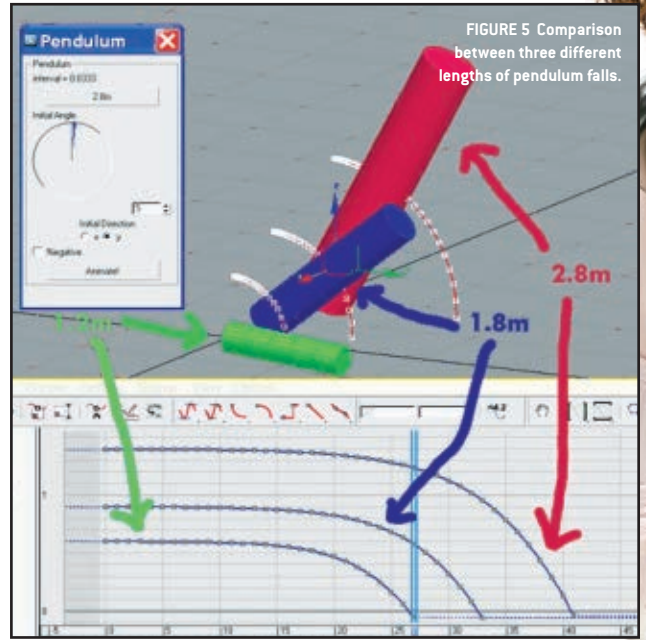
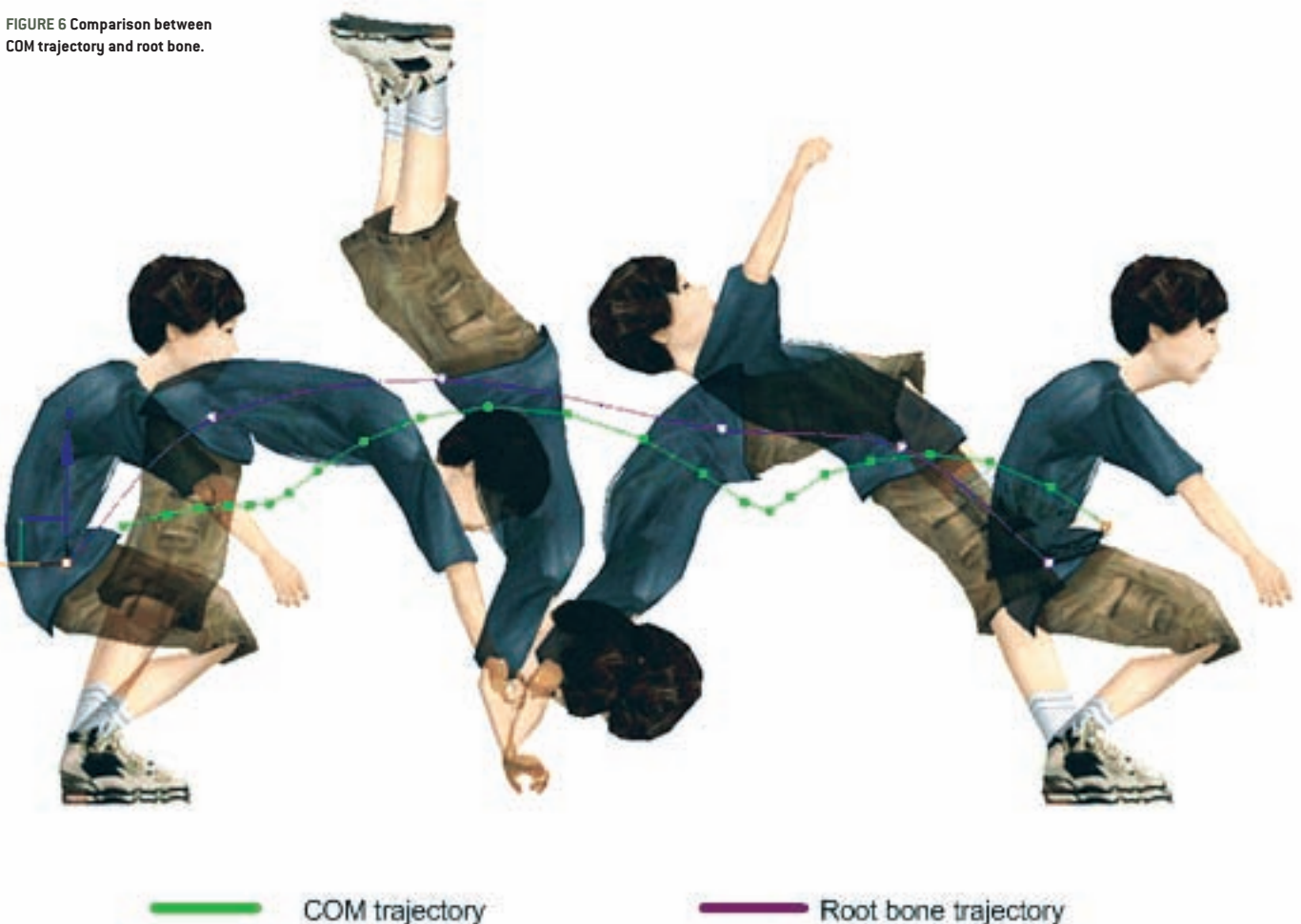


FIGURE 5 Comparison between three different lengths of pendulum falls.

FIGURE 6 Comparison between COM trajectory and root bone.



# CENTER OF MASS

FIGURE 9 Comparison animations with different centers of rotation.

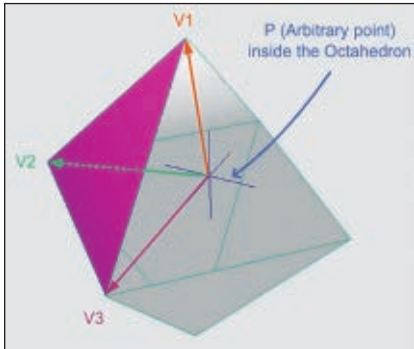


FIGURE 7 Calculating the mass of an octahedron.

certain actions using motion capture with a wire as well. When you animate a character, always remember the COM is not equal to the root bone.

## MASS APPEAL

» The COM in 3D is not something you can manipulate, so it's important to understand how the COM flow determines character movements.

Keep in mind that the human body is not a simple solid object, and is a compound structure with many joints and muscles. Because of this, human movement is complex and may not be something you can simulate precisely. But

if you use a simplified physics model (like the inverted pendulum) it will give you a good base to start with. Animators can use this base and modify it accordingly.

I hope the info I provided can give some of you a new perspective on how to create and analyze character animations. ☺

The images in this article are also available online at [www.kungfufactory.com/com](http://www.kungfufactory.com/com)

EIKO OBA has been the animation supervisor at Kung Fu Factory for the past seven years. Her most recent credits include UFC UNDISPUTED 2009. Email her at [eiko.oba@kungfufactory.com](mailto:eiko.oba@kungfufactory.com).

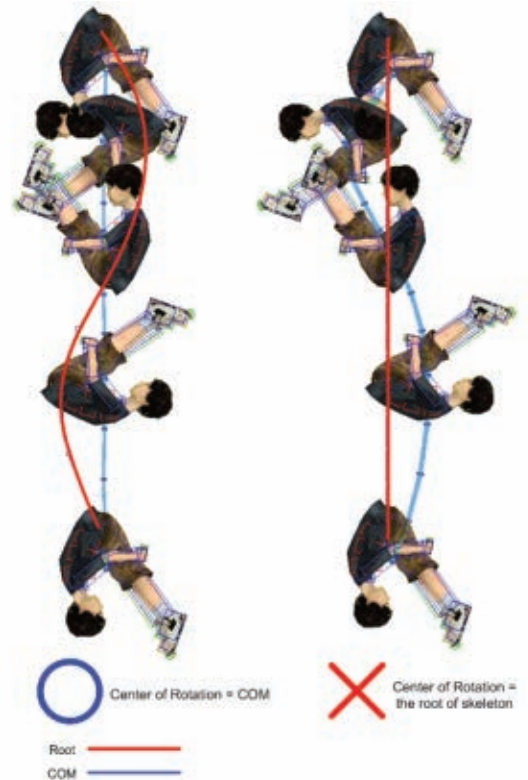
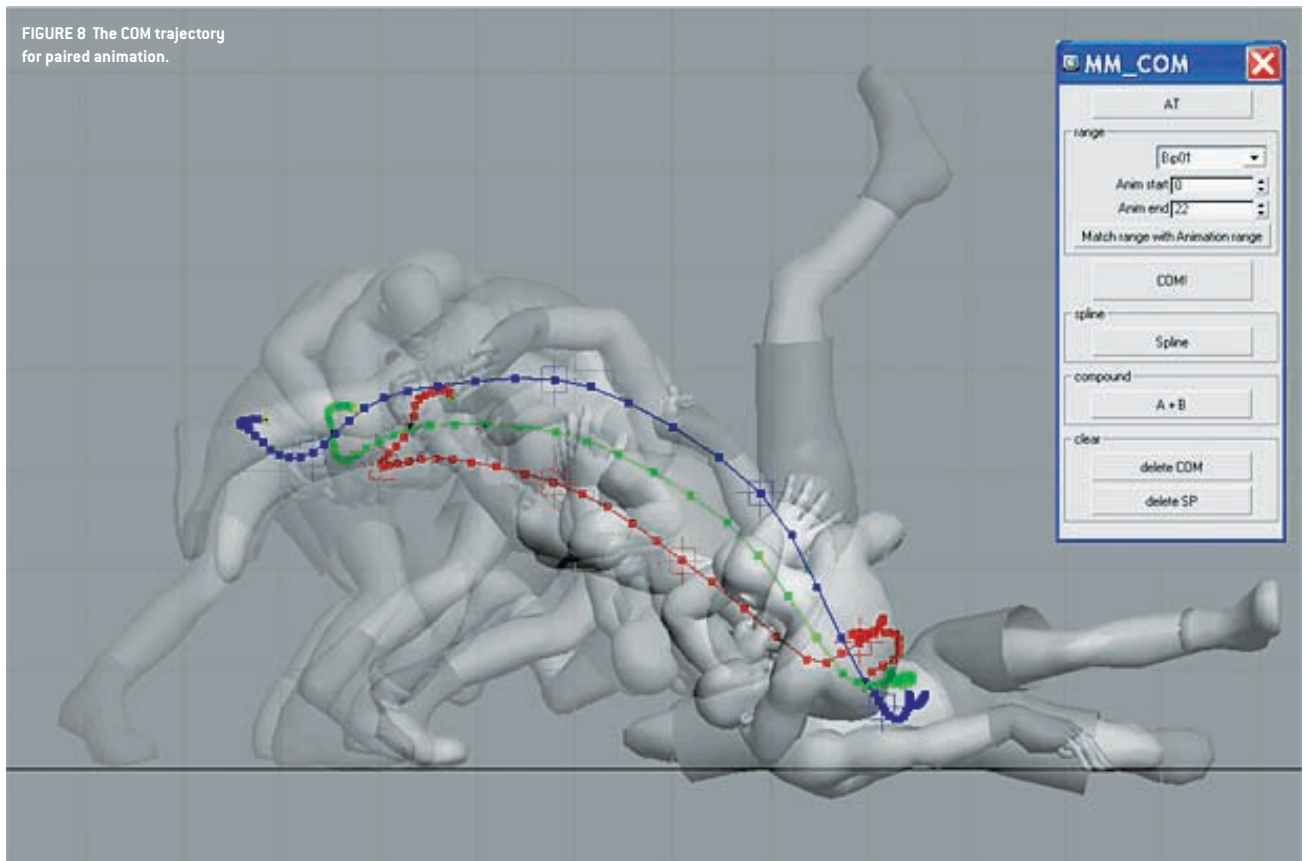


FIGURE 8 The COM trajectory for paired animation.



— Defender COM      — Attacker COM      — Compound COM

## GET YOUR GAME ON! EVERYTHING YOU NEED TO KNOW TO GET INTO THE GAME INDUSTRY!

- News and Features for students and educators
- Getting Started section – an invaluable how-to guide
- Message Boards



### DIGITAL COUNSELOR

I'll match your interests and goals with the right game related programs and schools from around the world.

[www.gamecareerguide.com](http://www.gamecareerguide.com)



DOWNLOAD AND PLAY THE LATEST STUDENT GAMES!

Download your FREE digital edition of the 2009 game developer Game Career Guide online at:  
[www.gamecareerguide.com](http://www.gamecareerguide.com)



PROGRAMMED BY THE GAME DEVELOPERS CONFERENCE IN ASSOCIATION WITH GAMECAREERGUIDE.COM

## BREAK INTO THE GAME INDUSTRY TODAY!

Don't Miss the Opportunity to Network and Learn From Game Industry Professionals.

# GDC 10

March 13, 2010

# GDC Canada

May 7, 2010

WHO KNOWS WHERE WE MIGHT BE NEXT - STAY TUNED.

[www.gamecareerseminar.com](http://www.gamecareerseminar.com)



# UNITY TECHNOLOGIES' UNITY 2.6

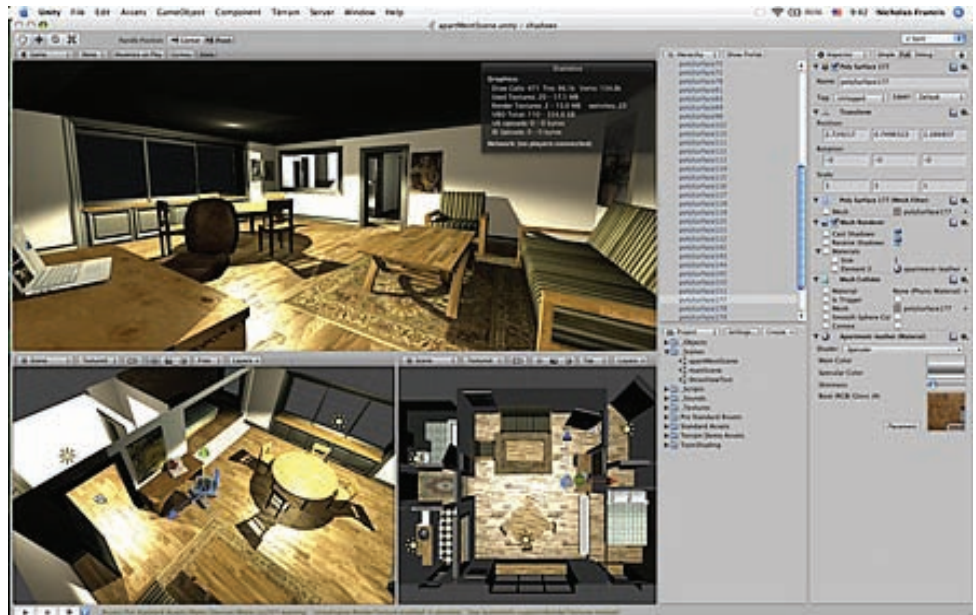
REVIEW BY CHRIS DELEON

THERE ARE CRITICAL ELEMENTS OF nearly every 3D game engine that most programmers would rather not have to reinvent for each new title: the level editor, scripting language, asset importation, rendering pipeline, and physics are all high on this list. Although engines exist to solve these problems, the free options are either incomplete, unstable, or a hassle to use, while the commercial options come with hefty price tags and strict limits on terms of distribution.

Unity offers a cure to the madness. After the important changes made in the past couple of releases—among them full support for PC development and several performance-boosting features—the few concerns that used to scare away some developers who were new to Unity have thankfully been resolved.

## EASIER BETTER FASTER

» I first heard about Unity when I tried to install the indie game RAPTOR SAFARI by Flashbang Studios. The web site asked whether I wanted the Unity plug-in, and a moment later, I was shocked to see a game taking place in sprawling 3D, using intensive physics, in my web browser. This wasn't the web game experience that I had come



Even more impressive than what Unity makes possible, is the developer tool set that makes it possible. Unity is a full-featured development environment, created to improve the efficiency of every member on a team. It seamlessly ties together asset integration, level design, and gameplay programming. Those on the business end will be pleased to know that the standard

instantly within the editor, at any time. Designers can even tweak variables within the editor while the game is running. The ability to instantly test gameplay streamlines programmer iterations by trivially verifying any feature in any scenario between changes. Artists can import native formats directly into Unity—from Photoshop, Maya, Max, and Blender—which automatically

download or CD distribution, ready to run full screen on Mac or PC. At this point there are even license options and development support for both iPhone and Wii development.

## LEARNING CURVE

» Learning Unity turned out to be much easier and faster than I expected. There are a ton of well-written and fully illustrated tutorials for Unity, and sites like the unofficial learnmesilly.com proved quite helpful.

As with any high-powered software tool, even though there are hundreds of menu options and keyboard shortcuts, at the core there are a handful of actions that happen dozens of times a minute, making up a majority of the input.

For basic navigation and object placement, the mouse works in tandem with Q (pan view), W (move selection), E (rotate selection), and R (scale selection). Alt+Mouse

Even though Unity's scripting engine is extremely powerful, any game engine that utilizes scripting for core game play risks additional overhead that can impair performance if developers are careless with the algorithms used.

to expect from my years of Flash development. This was the sort of project that I used to consider, with suitable caution regarding complexity, while I still worked primarily on downloadable games.

edition of Unity is available for free, while the Unity Pro version with additional rendering features is very reasonably priced.

At the heart of Unity's design process is the ability to play

update when edited. Even the player gets a good deal, since in addition to exporting a format for hassle-free web play using the free Unity plug-in, the development environment can export a standalone executable for

rotates the camera. Command+P toggles between in-editor play and editing. Command+Del deletes the current selection, to avoid the chance of accidental deletion. The F key instantly focuses all views on the currently selected object.

My favorite command though is Command+Shift, which makes any shortcut less than two keystrokes away. When I was starting out I liked this method because I didn't yet know where to find certain things in the menus, and as I've gained practice I've found that it's often faster than navigating menus by mouse.

For programmers, the best news is that Unity scripting is done in either C# or JavaScript, two very common object-oriented programming languages. The editor inherently favors the component approach to programming which maximizes reusability. Every public variable declared in script is exposed via the inspector for designers to tweak, tune, or hook up as they see fit. The documentation for the API is also extremely thorough. Did I mention that there's no need to rewrite 3D rendering, physics, file format importation, a level editor, or your own script parsing for the thousandth time? This enables every programmer to be a gameplay programmer—which if you ask me, is the fun part.

## HIGHLIGHTS

» There are a number of key concepts central to using Unity. Understanding these concepts is essential to getting the most out of the tool, so I'll offer a few points here to hopefully spare you the time that I invested into wrapping my head around them.

Everything in a Unity scene is a GameObject with components attached. Lights, cameras, players, planes, and models are all the same sort of neutral object (a "GameObject"), with the only difference being that they have different components added for light, camera, input, or other behaviors. Creating any of those objects is the same as creating an empty GameObject and then adding the related component.

Different levels within the same Unity project are created as different scenes. This includes menus, which are simply different scenes composed mostly of GUI components.

Naming GameObjects in the hierarchy is critical to staying organized. Objects can be grouped by making multiple GameObjects the children of a common empty GameObject parent. Objects are made children of one another by dragging the name of the child-to-be onto the parent-to-be in the Hierarchy display.

Unity centralizes design decisions—maximizing reusability and ease of maintenance—through prefabs. There is an important distinction between objects that are instances of Prefabs (archetypes, classes, shared properties) and instances that are broken from prefabs. Proper use of this feature can save a great deal of time and work by eliminating the need to copy and paste, or to manually propagate changes to multiple instances of the same object.

Drag and drop is widely supported in the interface, to a much greater extent than I'm used to, and this took some practice to become acquainted with. If there's a GameObject variable exposed for a script, the name of a GameObject can be dragged directly onto that slot from the other menus. Applying a Skybox from the provided assets was as easy as dragging the name of that material onto the spot for Skybox Material under Edit Render Settings. Assigning a GUIText object to a script to display ammo count is as simple as dragging the GUIText object's name from the hierarchy onto the slot exposed in the inspector for that script.

Variables changed during test play will reset to their original values once testing stops. There's an option in Unity's settings to tint the entire interface while temporary changes are being made. I recommend changing the default gray to a tint of blue, green, or red—this will look a bit unsightly, but can save time by preventing the loss of edits made while the game is running quietly in the background.

## Unity Technologies UNITY 2.6

### STATS

795 Folsom St. Suite 1059  
San Francisco, CA 94107  
<http://unity3d.com>

### PRICE

Unity: FREE  
Unity Pro: \$1499

### SYSTEM REQUIREMENTS

Windows XP SP2 or later; Mac OS X "Tiger" 10.4 or later, Graphics card with 64 MB of RAM and 4 texture units.

### PROS

- 1 The indie version of Unity is available for free—you can start using it today!
- 2 Editing while the game runs inside the editor offers an unprecedented level of control over tweaking and experimentation.
- 3 Unity's natural integration with outside tools—from native support for many audio/image formats to the newly improved tie-in to nearly any version control system—means that you can focus on learning Unity, instead of relearning aspects of game creation that you already know.

### CONS

- 1 Requires web plug-in for online audience.
- 2 As with any packaged engine and tool set, the ease of use and power inevitably comes at some compromise to the possibility for highest performance or customization with ambitious projects.
- 3 Unity is not the sort of environment that can be easily learned through experimentation alone. Because Unity's interfaces are designed to optimize development efficiency for experienced users, first time users will need to watch tutorial videos and read through forums to take full advantage of its power.

The terrain editor in Unity, while useful, creates new terrain that is enormous beyond practicality. Immediately after importing it, I'd suggest scaling it to a small fraction of its default size. Terrain also has a strange relationship to mouse selection which is helpful in practice, though confusing while learning: Terrain cannot be selected

by mouse in the display views, only in the hierarchy, and while selected, no other objects can be selected by a mouse click in the display views. Once you get the hang of it be sure to check out the free collection of terrain assets that are available for download from the Unity web site.

## NEW TO 2.6

» Even though Unity's scripting engine is extremely powerful, any game engine that utilizes scripting for core gameplay risks additional overhead that can impair performance if developers are careless with the algorithms used. One of the latest features, new to version 2.6, addresses this worry with a real-time performance profiler, enabling developers to diagnose which scripted processes are the most important to target for optimization.

There's a new animation editor that enables any property to be tweaked over time. This makes new kinds of lighting tricks, obstacles, and effects possible without needing to leave the editor environment. The improved animation system also makes it possible to initiate scripts from particular frames of animation. That feature makes it possible, for example, to guarantee that the damage or particle effect for an attack begins during precisely the frame of animation intended, rather than when the animation starts or after a separately tracked timer.

Unity's terrain engine for 2.6 now supports streaming. This enables a larger area to be created and populated without the burden such an area places on download time, memory, and processing time. Although there have been a number of performance enhancements made to the latest version of Unity, the gains from this change are likely the most dramatic.

Version control integration has also been improved for Unity 2.6. Unity now plays nicely with Subversion, Perforce, Bazaar, and other version control solutions. Whether you're integrating assets from others on the team or working alone while keeping careful backups, Unity is now better prepared than ever to work with



you and your team to protect and manage your assets during development.

Although 2.6 represents an incremental move forward, further polishing Unity's interface and features, the previous jump to Unity 2.5 was a major shift on many important fronts. Naturally all improvements from the previous version are at the heart of 2.6—if the last build you used predated 2.5, some of the key features you'll appreciate include the PC/Windows editor, improved object manipulation, and 3ds Max file support.

## PLUG-IN WOES

» There is one downside to Unity. While the company claims an install base of 16.5 million, players may not always have the web player plug-in installed. From a combination of security fears, discomfort with the unknown, and laziness, people can be reluctant to install plug-ins that didn't come with their computer. Fortunately, exporting a downloadable standalone player for PC or Mac removes the plug-in problem in exchange for the traditional distribution methods of download or CD. Although I'd prefer if we all went a bit out of our way to educate consumers about the benefits, safety, and simplicity of installing the Unity plug-in.

## A SMART INVESTMENT

» Although learning Unity isn't instantaneous, and the environment isn't entirely self-explanatory, it is possible to become proficient with the engine in a relatively short period of time. Compared to the amount of time required to get momentum versus what you can become capable of when using it, the payoff is remarkable.

I recently attended TIGJam 2009 in Arizona. Perhaps half of the indie developers present—many of whom weren't programmers by trade—were using Unity to pull together innovative 3D game demos over the weekend. In many cases I saw artists and designers working on video game projects that even dedicated programmers would find challenging.

If I'm sounding a bit evangelical, please note that I'm just another indie developer. This is my cry for help. I'm eager to see us all start changing the idea that Internet video games are limited to "match-three" clones, Checkers, and low-fi retro/nostalgia appeals. Since RAPTOR SAFARI, I've seen very few other groundbreaking projects online utilizing Unity, likely due to misconceptions about its platform support (now available for both PC and Mac), price (the indie version is free), or time required to learn (easy for programmers that already know either C# or JavaScript, and easy for designers and artists that have used other level editing environments). I hope that I've succeeded in clearing up these impressions, and provided the clarity needed to encourage you to give Unity 2.6 a shot.

**CHRIS DELEON** is a video game developer working primarily with iPhone and online video games. He is the writer and instructor behind the free lessons in video game development available at [www.gamedevlessons.com](http://www.gamedevlessons.com), and his experimental gameplay work is available at [www.interactionartist.com](http://www.interactionartist.com). His most recent work is *ALICE IN BOMBERLAND*, available now for iPhone and iPod Touch.

## product news

### WWISE 2009.3 RELEASED

**AUDIOKINETIC**

[www.audiokinetic.com](http://www.audiokinetic.com)

Audiokinetic released the latest version of Wwise, its audio pipeline solution. Wwise 2009.3 introduces new features and enhancements, including access to McDSP plug-ins. Audiokinetic also unveiled a new pricing structure that will allow customers to save up to 20 percent on purchases with the company's product families.

Wwise 2009.3 has a number of new features and enhancements including support for McDSP effects ML1 and FutzBox, peak meters on limiter effects, major memory optimizations for Vorbis, a new and improved file packager, and profiler data statistics.

### FREE EDUCATIONAL LICENSE FOR CRYENGINE 3

**CRYTEK**

[www.mycryengine.com](http://www.mycryengine.com)

Crytek announced that CryENGINE 3 is now available to educational institutions free of charge. For courses and research projects in game development, 3D graphics, simulation, architecture, animation, film, and design, CryENGINE 3 offers a development platform with scalable computation, graphics, AI, and physics for Xbox 360, PlayStation 3, and DX9/DX10. It provides the complete game engine to create next generation games, and includes the redesigned CryENGINE 3 Sandbox level editor, a "What you see is what you play" (WYSIWYP) tool designed by and for professional developers.

### HAVOK INDEPENDENT DEVELOPER PROGRAM ANNOUNCED

**HAVOK**

[www.havok.com](http://www.havok.com)

Independent developers now have a new way to use Havok

middleware in commercial projects, with the debut of the Intel-owned company's Independent Developer Program, an annual licensing arrangement.

Rather than licensing each middleware component on a per-case basis, the Independent Developer Program is an annual licensing arrangement that confers access to all of Havok's middleware products including Havok Animation, Havok AI, Havok Behavior, Havok Cloth, Havok Destruction, and Havok Physics. The suite supports multiplatform development.

Since last year, indie PC developers have been able to freely download the Havok Physics and Animation from Intel's web site. That offer remains intact.

### UNREAL DEVELOPMENT KIT NOW AVAILABLE FOR FREE

**EPIC GAMES**

[www.udk.com](http://www.udk.com)

Epic Games announced the launch of the Unreal Development Kit (UDK), a free edition of Unreal Engine 3 that provides community access to the 3D game engine technology. There is no charge for noncommercial or educational use of UDK and Epic Games will release ongoing, upgraded builds of UDK for free. Commercial terms have been structured so that independent developers, start-up firms and seasoned professionals can use UDK with minimal financial barrier from concept to deployment. Unlike mods, games created with UDK provide a standalone experience, with no additional software requirements.

The UDK offers immediate access to Unreal Engine 3 along with a suite of Unreal Editor tools. These include Unreal Content Browser for browsing, searching and organizing game assets with a collaborative metadata tagging system, the UnrealScript object-oriented programming language, and Unreal Kismet, a visual scripting system that enables rapid

prototyping, Unreal Matinee for building in-game cinematics, and Unreal Cascade, a particle physics and environmental effects editor that aids the creation of fire, fog, explosions, and other visuals.

The UDK also includes a NVIDIA PhysX-powered physics system, Unreal Lightmass, a global illumination system, as well as AnimSet Viewer and AnimTree Editor, which give animators control over muscle and bone movement. Unreal Engine 3 also features technology integrations with middleware tools including SpeedTree, Bink Video, and FaceFX.

Support for UDK includes over 200 pages of newly unlocked documentation at the Unreal Developer Network (<http://udn.epicgames.com>), dedicated forums ([www.udk.com/forums](http://www.udk.com/forums)), as well as other resources available through the UDK web site. 3D Buzz also hosts hundreds of free video tutorials for using Unreal Engine 3 technology at [www.3dbuzz.com](http://www.3dbuzz.com).

### SHIVA EDITOR 1.8

**STONETRIP**

[www.stonetrip.com](http://www.stonetrip.com)

Stonetrip, the designer of cross-platform 3D game development tools for Web, PC, Mac and iPhone, has announced that version 1.8 of its ShiVa Editor is now available. ShiVa Editor allows developers to publish games and applications with no limitations and no compulsory content [e.g. splash screens]. The ShiVa editor comes with a free engine SDK for Windows, Mac, Linux and iPhone plus an integrated server included within the player to create multiplayer games without the need for an external server.

ShiVa 1.8 features depth of field, velocity blur, cascaded shadow maps algorithm, infinite ocean simulation with advanced shading system, dynamic textures, video streaming, as well as Voice-Over-IP, DWF import and performance improvements.



# GDC Canada

**Learn. Network. Inspire.**

Game Developers Conference® Canada

**May 6-7, 2010**

Vancouver Convention Centre | Vancouver, BC

Visit [www.GDC-Canada.com](http://www.GDC-Canada.com) for more information





# BRICK BY BRICK

VOXEL TRACING METHODS ON CURRENT AND UPCOMING GPUS



An example of voxel raytracing from AMD's 2008 Cinema 2.0 presentation.

VOXEL REPRESENTATIONS ARE UBIQUITOUS IN HIGH end film rendering and scientific visualization, but are traditionally far too demanding in computation and storage costs to be used in games. This is set to change. In this article, I will outline the case for an upcoming transition from polygon rasterization to voxel tracing as the dominant rendering paradigm as we move to complex scenes on future hardware, and some approaches to practical voxel tracing on current graphics processors with an eye toward complete voxel engines for upcoming next generation platforms.

Polygon representations are very efficient for modelling smooth surfaces, but scale poorly with very high scene complexity and fractal dimension. Level of Detail techniques are required to scale to highly complex scenes, and this is equally true of both polygon and voxel representations. In the limit case, where we desire a near lossless representation of a massive scene, complex geometry consisting of numerous surfaces must ultimately be simplified down to a single triangle, which is considerably more complex than a single voxel. Vegetation is the typical difficult case, as it has a high fractal dimension (ratio of surface area

to volume), so a near lossless representation will require roughly equivalent numbers of polygons or voxels. On the extreme end of this continuum we have fully volumetric phenomena such as clouds, smoke, fire, etc., for which voxels are a natural fit. So once voxel methods become performant, they have the considerable advantage of a single unified representation. We can see an analogy within the history of 2D graphics, where at one point image-based graphics were too expensive and vector graphics dominated. Voxel graphics are the extension of image techniques into full 3D, and will eventually have all the equivalent advantages. The final consideration is that modern GPUs are no longer advancing in polygon throughput, even while their general FLOPs and bandwidth grow exponentially. Current high end cards would choke on scenes with hundreds of millions of triangles, but can easily trace a voxelized version of the same scene in real time today.

## VOXEL TRACING APPROACHES

» Voxels are just a data representation, and like triangles are amenable to rasterization or tracing. Voxel rasterization through splatting or projection

is certainly a viable technique, but it is not very interesting. The key challenge of high quality rendering is illumination, and tracing is a simple yet powerful framework for complex illumination effects. Even if primary ray generation were free, secondary rays grossly outnumber primary rays as we move toward high quality illumination.

There are numerous spatial acceleration structures that ray tracers can use: octrees, axis aligned bounding box (AABB) trees, and bounding volume hierarchies (BVH), to name a few. Any tree structure that normally stores triangles in its nodes can just as easily store voxels. For GPU implementation simplicity and regularity are desirable, so I have focused on variants of regular trees, or M-trees, which are just trees of uniform sized nodes with branching dimension M. With M=2, we have our familiar octree. Higher M counts provide a memory versus query cost tradeoff, allowing shallower trees that can be traversed faster, at the expense of less memory efficiency. Voxels are then stored in each node, which can be aggregated into regular "bricks" of BxBxB voxels to further reduce the tree depth. This M-tree of B<sup>3</sup> bricks structure is common in research



literature—for the most recent treatment see Cyril Crassin's GigaVoxels project (in Resources, pg. 40) which describes a similar architecture in detail.

Jon Olick has developed another variant within the general octree voxel scheme which is described in his SIGGRAPH 2008 presentation: "current and next-generation parallelism in games." Olick's system is not brick-based, which makes some very different tradeoffs versus the brick-based schemes such as GigaVoxels or my own (which I will describe in the following pages). Storing a single voxel per node can actually be more net memory efficient with compression and it permits fast specialized traversal techniques, but it has a significant disadvantage when filtering is considered, as we will discuss. Olick also uses a more complex irregular octree, which can be offline optimized to better fit the data, at the expense of more complex traversal and dynamic updates. In my own research, I have independently taken a route more similar to the GigaVoxels approach, influenced by the same preceding research in regular M-tree voxel brick systems.

## DATA STRUCTURES

» There are numerous ways to implement a regular M-tree B-brick voxel data structure. I use a structure that explicitly distinguishes between inter nodes (which have MxMxM child pointers), and leaf nodes (which do not)—primarily to save memory. All nodes are associated with a BxBxB brick of voxels to permit mip mapping. I mainly take a structure of arrays approach with common indices. The memory-dominant structure is the array of voxel bricks, followed by the array of child pointer bricks, but I also use an array of parent pointers and an array of compressed accessory node data (position, depth, score, and some flags), amounting to some 30 percent excess over the voxel brick data. In addition, even though M is usually 4, I always store an extra array of the 2x2x2 direct child pointers for each node, as this basic octree structure is easier to manipulate during updates. The wider child pointer array can easily be generated from the base octree. The pointers are all 32-bit indices, but as there are at most a few million nodes some of those bits are used to store additional flags. Reasonable values of M and B are 4 and 8, which means a voxel brick stores 512 voxels and a pointer brick stores 64 pointers. This scheme allows a fast descent through the structure and doesn't use too much extra memory. Specifically, the wasted memory is directly proportional to the B value, as larger bricks tend to waste more on padding voxels that map to empty space or interiors. Crassin et al's GigaVoxels system takes a different storage approach with nodes storing either an indirect pointer to a voxel brick or a constant color value for the whole node. Refer to their papers for more details.

## FILTERING AND CONE TRACING

» The most significant advantage of storing a brick of BxBxB voxels per node is that it permits fast high quality filtering of voxels using the GPU's native texture filtering hardware. Manually performing a 3D linear filter would require eight memory fetches and many dozens

of instructions, so replacing this with a single native texture fetch is a huge gain. Furthermore, smooth linear filtering naturally permits direct alpha accumulation, allowing the approximate tracing of entire cones instead of rays.

The exact intersection of a cone with a voxel is complex, but a cone can be reasonably approximated by a series of spheres, which in turn can be reasonably approximated by a series of box filters—which themselves are just single 3D linear texture filters. This successive approximation is not nearly as bad as it may first sound for two reasons: Firstly the cones are typically quite narrow, and secondly, boxish frustums are just about as reasonable a choice as cones for modelling the flow of light projecting onto a pixel.

Tracing cones can be far more efficient than tracing rays. Rays are discrete approximations, and tracing them requires the accumulation of dozens of rays to converge correctly and reduce aliasing. Effects such as soft shadows, anti-aliasing, anisotropic filtering, depth of field, proper blurry reflections, and even ambient occlusion can be done with just a single cone trace or a small number, and global illumination effects can be done with dramatically fewer cones than rays. Adaptive subdivision in screen or ray space can reduce the required rays for some effects, but the advantage is dependent on the scene complexity, and can break down for complex scenes (jungles, grass, and so forth). Overall, cones are vastly more efficient, and are not significantly slower to trace for the octree brick schemes we are discussing—so we essentially get all these otherwise difficult effects for free.

## TRAVERSAL STRATEGY

» Efficient GPU ray traversal algorithms have been developed for polygon ray tracers, and we can use any of these as a starting point. The main difference is in the inner leaf node intersection: instead of intersecting with a list of triangles, we intersect with a brick of voxels, which just involves stepping through the brick and accumulating the voxel samples—simpler and faster than intersecting rays with triangles. Timo Alia et al published an excellent comparison of various traversal methods on modern (nvidia) GPUs: "Understanding the Efficiency of Ray Traversal on GPUs" (see Resources). They compare triangle ray tracers using a Bounding Volume Hierarchy (BVH) representation, but their results are still relevant for any ray traversal methods. Ray traversal could be implemented in DirectX Compute Shaders or OpenCL and is generalizable to current AMD GPUs, but my implementation and all the others discussed here are implemented in Nvidia's CUDA, and some of the performance considerations may be somewhat platform specific. For brevity's sake, I am going to assume some familiarity with the CUDA architecture—the latest documentation can be found on Nvidia's site.

There are two main performance issues with ray traversal on a GPU: stack structures and divergent branching. Straightforward ray traversal requires a stack data structure per ray, which is difficult on a GPU due to the very limited (or nonexistent) fast local memory. There are a couple of options with Nvidia's CUDA architecture: shared memory (on-chip), or global (off-chip) memory (there is also local, but it's the same as global for our intents). Alia et al get surprisingly good performance by storing the stack in global memory, even though this requires constantly writing and reading to off-chip memory (they are not memory bound). My octree voxel tracing method avoids the need for a stack, as does the GigaVoxels technique, instead using a stackless restart method, as we will see. The other performance pitfall is thread divergence, which is the real killer. Both sides of a divergent branch must be evaluated, which can be quite wasteful due to the SIMD width (effectively 16-wide for GT200 CUDA warps). There are numerous variations, but the stackless while-while traversal used in GigaVoxels can be efficient for M-tree brick structures:

```
while-while restart trace()
  while ray not terminated
    set current node to root node
    while current node depth is less than target LOD depth
      traverse down to the next node
    while ray is within node's voxel brick
      sample current voxel within brick and accumulate
```

The restart traversal method avoids the need for a stack and any extra logic to walk across neighbor nodes. It does redundantly traverse from the root down to the relevant leaves over and over, which is wasteful, but large bricks and a shallow tree can compensate for this. For example, with M=4 and B=16, it will take less than ten steps to jump down to a leaf node, and this time is



amortized over about a dozen voxel samples. Of course, such large voxel bricks waste memory and increase the cost of streaming and dynamic data generation. For small voxel bricks, a neighbor-hopping traversal strategy may be better:

```
if-while stack trace()
  set current node to root node
  while ray not terminated
    if current node depth is less than target LOD depth
      traverse down to the next node
    while ray is within node's voxel brick
      sample current voxel within brick and accumulate
      traverse to next neighbor node
```

This neighbor hopping traversal requires some extra memory and logic, but reduces the time spent drilling down from the root to the relevant leaf node. However, we should consider the relevant cost of some of these pseudocode operations. Traversing down a child node is extremely simple in a regular octree, and can be done in just a handful of vector instructions. Stepping and sampling within a brick is likewise very simple and fast. Traversing to a neighbor node is more complicated, but can be accelerated with a lookup table.

As an alternative to neighbor-hopping, further optimizations to the restart scheme are possible by reducing the steps to jump from the root to a leaf, leading to a very simple divergence-free stepping:

```
branchless stackless restart trace()
  while ray not terminated
    set current node to root node
    warp from root node to target LOD leaf node
    sample current voxel within brick and accumulate
```

Obviously I'm obfuscating a bunch of details in "warp from root node to target LOD leaf node," but this can be done with a small deterministic number of steps in a shallow tree with a known maximum height. This last scheme can be faster with small bricks and is less sensitive to divergent ray issues, at the cost of more stepping logic per voxel sample on average.

## PERFORMANCE AND TWO LARGE OPTIMIZATIONS

» A typical shader strategy is to allocate a thread per pixel, and thus a thread per ray. In CUDA, these are then batched up into user-sized blocks, which are dispatched to the GPU's cores (which Nvidia calls multiprocessors). A straightforward equal allocation of image tiles to blocks works well for homogeneous tasks, such as shading, but performs poorly for heterogeneous tasks such as ray tracing where thread execution times vary widely. Aila et al propose a simple, effective solution: persistent threads. Instead of allocating enough thread blocks to tile the entire image, allocate about one thread block per multiprocessor, and implement dynamic work stealing using atomic operations. This way no multiprocessors get starved. They found this almost doubles performance on average, and I found this to be just as true for the octree traversal methods.


The cone tracing strategy outlined above permits another large performance gain through simple hierarchical tracing. With hierarchical tracing, the image is created in a pyramidal coarse-to-fine fashion, starting with a low resolution version, say 64x64. The image is traced until rays accumulate an alpha value above some epsilon threshold (i.e., encounter anything), after which they terminate early and output depth. When the next level in the pyramid is traced, rays can start at the depth output from the previous level instead of starting at the camera, vastly reducing the distance to trace for the finer levels of the pyramid. This is fairly straightforward to implement in a cone tracing system using linear filtering, and it is possible but considerably more difficult for discrete ray tracers.

With these optimizations, I found that ray traversal of a 720p test scene with several hundred million voxels on an 8800 GT can take anywhere from 5–16ms. This is without any dynamic shading, just primary ray casting. Somewhat surprisingly, a prime source of variability is the anisotropy, because at high glancing angles rays spend far more time skimming leaf nodes and get less benefit from hierarchical tracing. Geometric complexity is not as much of an issue (as there are always the same high number of voxels used to represent any scene). This is quite fast—

not as fast as rasterizing a typical game's scene on the same hardware, but we are talking about orders of magnitude more geometric complexity than typical game scenes, so it's far faster than rasterization in that sense—and could be significantly faster with more optimizations. These results are comparable to GigaVoxel's published results (30–90 fps on a variety of scenes for a 512x512 window), and Jon Olick's 2008 SIGGRAPH demo (30–60 fps at 720p for a scene with a single complex model) on similar hardware. For comparison, these timings represent throughput rates ranging from 50–200 million rays/sec. For comparison, the highly optimized BVH triangle tracer of Alia et al achieves around 70–140 million rays/second on a GTX 285, which is about twice as fast as a 8800 GT, so voxel cone tracing can be faster than triangle tracing—on top of the previously mentioned huge benefits of tracing cones over rays.

## CONCLUSION

» Octree voxel tracing is already possible at reasonably high performance on video cards just one generation above the current consoles, so we can expect adequate performance to ray trace at considerably higher resolutions and or to trace significant dynamic secondary rays on the next generation consoles and upcoming high end GPUs. Memory cost can be a limiting factor and compression, active memory management, and streaming are all important for a complete system, which are beyond the scope of this article (but are tackled in detail in some of the referenced papers).

Voxel tracing can be combined with deferred shading strategies, or shading and secondary ray generation can even be integrated into uber-kernels, as in Nvidia's OptiX triangle ray tracing architecture. A particular challenge not addressed here is the issue of dynamic updates, which is a whole area of further research in this exciting realm of graphics. I've covered the basics of voxel octree data structure and traversal strategies, which are the core nuts and bolts of a ray (or cone) tracer, but that is just the tip of the iceberg. I expect that voxel tracing engines will give polygon rasterizers a run for their money in the years ahead, and eventually take over for most rendering tasks because of their power, scalability and generality. 

*JAKE CANNELL* most recently worked as a graphics programmer at Pandemic Studios, and has been creating graphics engines and demos at work and home for about ten years. He maintains a blog to speculate on such matters at [www.EnterTheSingularity.com](http://www.EnterTheSingularity.com).

## resources

**Cyril Crassin's GigaVoxels:** <http://artis.imag.fr/Publications/2009/CNLSE09/>

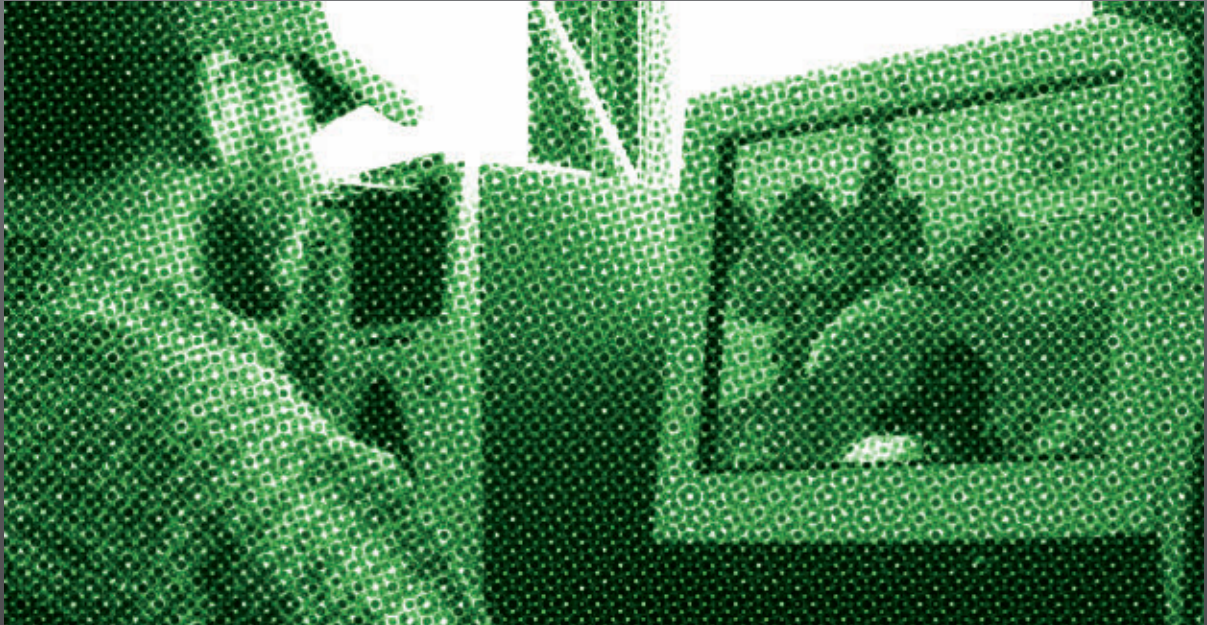
**Jon Olick's SIGGRAPH presentation on voxel octrees:** <http://s08.idav.ucdavis.edu/olick-current-and-next-generation-parallelism-in-games.pdf>

**Timo Alia et al: "Understanding the Efficiency of Ray Traversal on GPUs"** [www.tml.tkk.fi/ftimo/publications/aila2009hpg.paper.pdf](http://www.tml.tkk.fi/ftimo/publications/aila2009hpg.paper.pdf)



# GAMASUTRA

The Art & Business of Making Games



**THE LEADING PROFESSIONAL  
GAME SOURCE FOR JOB SEEKERS  
AND EMPLOYERS**

Access the Web's Largest Game Industry  
Resumé Database and Job Board

Take Control of Your Future Today!  
Log onto [www.gamasutra.com/jobs](http://www.gamasutra.com/jobs)

**THINK**<sup>™</sup>  
SERVICES  
A DIVISION OF UNITED BUSINESS MEDIA LLC



# TILIN'? STYLIN'!

## THE LOST ART OF VIDEO GAME MASONRY

**IN THE SHORT BUT EVENTFUL** history of game art, one of the oldest traditions is the craft of the tile. Environments built from repeatable sets of tiles (or more rarely, hexes and triangles) go back to the musty days of the Atari 2600, Colecovision, and the other 8 bit dinosaurs. In the last decade, with the triumph of 3D graphics, tiles have become less of an issue. Texture artists have never lost touch with the tradition, but most working artists today are more familiar with the ins and outs of sculpting than the arcana of the tile trade.

Nowadays, tiles are making a comeback. The rise of handheld and casual games has made tiling (a technique that evolved in hungrier days when memory was at a steep premium) interesting again. It's not just DS games that need tiles, though—even in the AAA world, the popularity of open world games has a lot of teams looking for ways to fill big spaces efficiently and cheaply without surrendering artistic control. As authoring becomes more complex and expensive, anything that wrings more use out of precious art time is worth a second look.

Tile-based authoring combines hand-crafted detail with the flexibility and efficiency of procedural content. The simplicity and immediacy of tiles also allows designers and programmers who can't manage hardcore 3d apps to put together slick, professional looking environments. So this month, we're going to revisit some of the lost secrets of the tile-makers' art.

### TILE IT UP!

» Tiles are modular units that can be combined to create environments. Oldschool 2D bitmaps are one example—but so

are complex models combining 3D geometry, textures, and gameplay markup. Full 3D tiles are of course much more expensive to create than simple textures, but the same basic design rules apply.

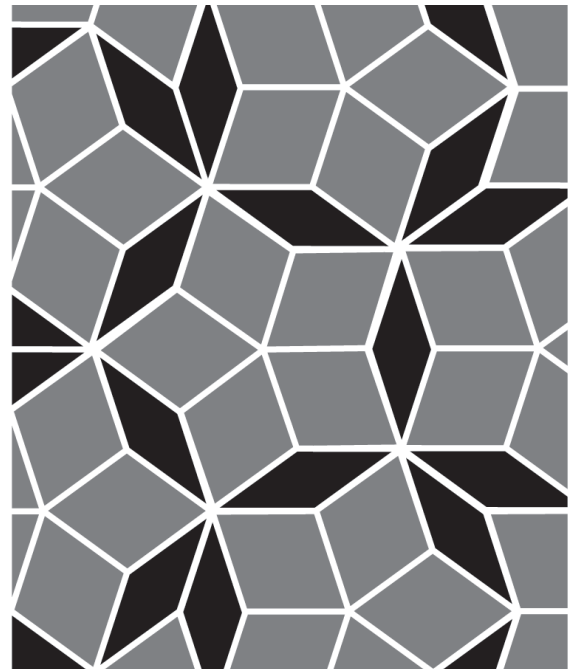
The essence of tiles is that they can be connected in regular fashion. The usual metaphor comes from board games like *Labyrinth* or *Carcassone*, in which you construct the game board by placing tiles with matching edges. It's important to recall, though, that our tiles don't have to be squares: Any collection of shapes that can be connected regularly can be used as an environment-building toolkit. Oldschool wargamers (and fans of *HeroScape*) know that hexes make great tiles. Equilateral triangular tiles are also frequently used, as these shapes can be repeated infinitely. There are in fact lots of shapes that could be repeated infinitely as tiles (as thousands of M.C. Escher prints on thousands of dorm room walls conclusively show).

It's rarely done in games, but it's also possible to create tile sets from more than one shape. In the traditional art world, though, it's common to find mixed tilings—you probably have a mixture of octagon and diamond tiles in a bathroom. Games don't typically need the intricate symmetries of the Alhambra (See Figure 1) but tile sets combining different shapes can be a very powerful tool for controlling visual rhythms. Regular tile sets are always vulnerable to monotony. Mixed sets and unusual shapes like the Penrose tiles (see Figure 2) can be useful because they avoid the mechanical regularity of more common grid.

If tiles can be almost any shape or combination of shapes, how can you hope to plan a decent tile set for practical use? And what kind



**FIGURE 1** Tiles from the Alhambra Palace in Granada, Spain. The Islamic prohibition on figures of people and animals led to a flowering of abstract, mathematically grounded tile art in the Moorish Kingdoms of Spain. Photo: Roberto Venturini.



**FIGURE 2** Penrose tiles consist of two shapes, which create patterns with a 5-fold symmetry—an interesting, if tricky to manage alternative to squares and hexagons. In this case, two rhombus shapes make tiling with a star pattern at the center. The markings on the edges indicate matching sides.

of rules could possibly apply to such a varied problem set? Luckily, the basic rules for building an environment out of tiles are pretty simple, and they apply equally to everything from plain old bitmaps to complex 3D models, and from simple squares to Penrose darts.

## LIFE ON THE EDGE

» The relationship between neighboring tiles is much more important than the contents of each tile individual tile. When great individual assets are juxtaposed awkwardly or too predictably, their individuality is undermined and the tiling becomes oppressive. If you want your environment to have any kind of flow, you need to think hard about how neighboring tiles relate. At least you can take comfort in the fact that edge matching works the same way for any shape tile. The rules for transitions between hexes, triangles, or even mismatched shapes are the same as for boring old squares.

The first, and most jarring problem is how tiles match across edges. Often, tile design is really more like edge design. Since the edges have to match, they're an inescapable visual element that has a strong influence on the final environment. They're also the key determinant for gameplay, dictating player routes and lines of sight. Edge matching also dictates whether tiles can be rotated when they're placed. For these reasons, planning a tile set really starts with planning sets of edges.

Since edges are so visually intrusive, you need to carefully balance the relative importance of the tile's unique content with the transition areas along the edges. Large transition areas disguise the exact shape of the tiles. Unfortunately, they also tend to be bland. Small transitions, on the other hand, emphasize the "tile-ness" of the tiles (see Figure 3). It's important to vary the size of that transition zone—if it's always the same size, it over-emphasizes the underlying rhythm of the tile set. It's particularly important that the center of interest in every tile is not rigidly aligned to the tile.

A useful trick when thinking about these transition zones is to imagine the tiles as if they were actually subdivided into triangles from a central point, as in Figure 3. These triangles show the influence of the different edges. Moving that central shared point off center lets you emphasize the influence of some edges and maximize the influence of others, which is very helpful for disguising the underlying tile grid.

The physical size of the tiles matters too: Larger tiles downplay the transitions, because the percentage of space you need to devote to edge management goes down and the amount of unique (and hopefully interesting!) content goes up accordingly. Unfortunately, larger tiles are more expensive to create, which means you'll probably have fewer of them to work with. Designers, in general, are happier with a more flexible set of small tiles they can play with—artists tend to love big, detailed tiles that are individual works of art.

## CORNER CASES

» Edges are visually the most important part of tile sets, but they don't tell the whole story. Diagonal relationships are a sticky design problem. Square and triangular tiles will be next to neighbors across their corners as well as their edges (Hexes don't have this problem to the same degree, because the other side of a hex transition is always a pair of edges rather than a new tile).

Corners are a nuisance. Unlike edges, they don't dictate placement—which means you can't design the corner of one tile with any knowledge of the tile diagonally opposite (see Figure 4). This leads to designs which emphasize the artificial tile boundaries at the expense of natural flow (as you can see in Figure 5).

If you compare the tile set from Figure 5 to that of Figure 4, you can see that tiles which include transitions inside, rather than limiting them simply to edges, are much more flexible. Edge-only tiles like those in Figure 4 are common in strategy games where the transition



**FIGURE 3** When the transition zone between tiles is very tight, the underlying grid of the tiles is emphasized (top) Longer transitions disguise the pattern better (bottom). Varying the size and placement of the contents avoid stiffness and repetition.

is purely cosmetic and the entire tile is going to have a single set of game properties. Unfortunately, edge transitions create a strong visual reminder of the underlying grid.

If your tiles don't have to double as spaces on a game board—if they are merely modular building blocks for a 3D environment construction kit, for instance, it's easier to use a scheme like that in Figure 5. By adding tiles which include both terrain types, you avoid both the corner matching problem and the rigid alignment to the grid. By adding a third tile type, you avoid the corner matching problem and you're no longer stuck with those pinched corners. Adding more variants to the transition set makes it much easier to disguise the underlying structure of the tiles.

## OFF THE GRID

» We love tiles because they're extremely efficient. Of course, to artistic eyes, they often seem stiff and constrained. No matter how interesting and different the content of the tiles might be, the regular spacing imposes a very obvious

rhythm that undermines the illusion of variety. Strict regularity is deadly if you're trying to create an organic feel—an outdoor landscape, a wandering cavern, or a medieval village. Even in a modern urban setting it's going to get pretty dull.

Combating the stiffness of a tile-based system demands a mixture of tactics. We've already pointed out that it's important not to lock the center of interest in your tiles rigidly to the physical center of the tile. Another key tactic is to vary the number and scale of key features (another trick every texture artist knows by heart). If your tiles contain the buildings of a medieval village, don't make each tile a single house: One might be cluster of outbuildings, a small shed up against the tile edge, a cottage with a grove of trees—anything but one more similarly sized house every X feet! As always, big tiles make this easier to pull off visually—if your designers and programmers don't mind the loss of layout options.

Another antidote is compound tiles, which cover the area of several

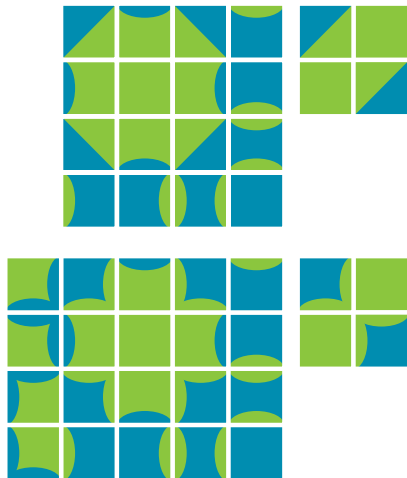


FIGURE 4 Two tile sets based on edge connections. Edge-based tiles are simple to create, but inflexible. They also create ambiguous situations across diagonal corners.



FIGURE 5 A transition tile allows edges with multiple edge types. The minimum set of tiles is larger, but the layout is much more flexible and diagonal connections are now clear.

tiles at once but don't have internal boundaries. Whether you build them as tiles with unique edge types or big single assets, compound tiles help shake off the tyranny of the tile grid while retaining the flexibility of modular construction. As you can see in Figure 6 it's especially helpful if the main features of the mega-tiles aren't rigidly aligned to the grid. Important landmarks and key gameplay locations tend to predominate



FIGURE 6 Compound tiles are an important way to break up the grid and add variety.

among compound tiles, but its important for visual variety to have a few big blocks that can be repeated as well. Outdoor features can also be compounds—a lake or a hill that would be too stiff and geometrical when built of regular pieces can be handled well as a compound.

### ROTATIONS

» If you design your tiles so they can be rotated, you'll get a lot more visual variety. Rotations add a lot of flexibility in layout and make good use of your hard work. If you don't design with rotation in mind, though, rotations can also undermine your illusions. Tiles with a strong internal structure are hard to rotate. If you want rotatable tiles, avoid things like baked-in shadows or strongly directional textures. Tile sets are a great application for technologies like normal maps and screen space ambient occlusion that respond gracefully from any viewing angle.

Even if you're not planning on rotating tiles, don't be a slave to the tile grid. In some environments, like dense cities or modern interiors, strict alignment is a feature rather than a drawback; but in fluid or organic settings avoiding alignment is important. Our eyes are particularly sensitive to angles in multiples of 15

degrees, and the regularity of a tile grid will tend to heighten the effect. Unless you really want to emphasize a mechanical or urban look you should go out of your way to avoid the giveaway angles.

### MORE TO COME

» A well-designed set of tiles is an extremely powerful tool, combining the artistic nuance and craft of hand created content with the flexibility to fill up large game spaces on the cheap. Like any strategy for content re-use, we're constantly struggling to find the right balance between visual variety and economy of effort. The right balance for your project has as much to do with external forces—memory budgets, time, and manpower—as it does with the eternal laws of art. But taking a little time to revisit the basics will pay off—and it'll help give you some appreciation for those old-school Pixel Pushers of yore. 🎮

**STEVE THEODORE** has been pushing pixels for more than a dozen years. His credits include *MECH COMMANDER*, *HALF-LIFE*, *TEAM FORTRESS*, *COUNTER-STRIKE*, and *HALO 3*. He's been a modeler, animator, and technical artist, as well as a frequent speaker at industry conferences. He's currently a consultant helping game studios perfect their art tools and pipelines.

ACCESS IT IN REAL TIME

# gamedeveloper

DIGITAL EDITION



*Game Developer* magazine's 6 month and 1 year Digital Edition subscriptions give you new issues delivered promptly, full searchable access to more than 50 back issues, plus downloadable PDF versions, easy access from any web browser, and more!

Subscribe today!

[HTTP://WWW.GDMAG.COM/DIGITAL](http://www.gdmag.com/digital)

**THINK**  
SERVICES  
A DIVISION OF UNITED BUSINESS MEDIA, LLC



# GET REAL

## FUN VERSUS REALISM

### HOW REALISTIC IS TOO REALISTIC?

One can ask the makers of *The Polar Express*, the animated Christmas film released just in time for Thanksgiving in 2004. Despite huge investments, a big-named director, and Tom Hanks providing the voice and mocap animation for several roles, the film struggled at the box office, getting swamped by another animated film, *The Incredibles*, released five days prior. Around this time, the concept of the Uncanny Valley entered the public mind.

The Uncanny Valley is a theory that most game artists (especially modelers and animators) are well aware of now, but it wasn't always the case. Around the same time *The Incredibles* was trouncing *Polar Express* in the box office, too many art directors believed real games made for gamers had to chase photorealism in order to be successful. *WORLD OF WARCRAFT* eschewed all of that for a cartoony appearance, and in doing so blew past all the competition and expectations. Three years later, *TEAM FORTRESS 2* would do the same for a shooter

lifelike visuals now, but it's also increasingly expensive and time-consuming to generate that content, and the end result is a screenshot that looks not all that different from competitors who are also chasing realism as an end goal. But realism isn't just a pitfall for artists—game designers also flirt with realism as a source of inspiration for their game mechanics, often with staggering implications to their game designs.

### THE REALISM TRAP

» The unwary designer can get into trouble by trying to follow realism too closely. Making a scene look realistic doesn't necessarily make it look more beautiful, fantastic, or intriguing. Similarly for designers, a game mechanic that is realistic doesn't necessarily make the game fun.

A common way that this makes itself evident inside our game designs is the rise of sandbox games. Once a mechanic largely limited to strict simulations, the success of *GRAND THEFT AUTO* has resulted in game designers trying to

walls. True, it's more realistic, but it's also more expensive to build and test that world.

And even if it weren't, sandbox gameplay may fight with other tenets of the design. For example, most players get confused and overwhelmed when told to find their own fun, and systems need to be devised to lead them to interesting activities. Compelling narratives are harder to tell, because designers lose control of the order and flow. Sometimes issues are more insidious: *BURNOUT: PARADISE*'s open world structure made it difficult for players to attempt to do the same race or challenge twice in a row, as many racing game players want.

Are sandboxes inherently bad? No—some of the finest games in the world are sandboxes. But injecting this level of realism into a game has very direct repercussions on the cost and design of the game that the designers must be mindful of.

### WHEN REALISM CREATES UNREALISTIC BEHAVIOR

» In the early days of *EVERQUEST*, it was not uncommon to stumble upon another player in the wild who was throwing himself off a short cliff over and over again while spewing gibberish indecipherable to passersby. Use-based advancement was to blame: While most of *EVERQUEST*'s advancement model was centered around a classic level-based advancement system, the non-combat skills like "Language" and "Safe Fall" advanced as the character performed in-game actions. Thus, our mysterious cliff diving tonguespeaker was someone whose character was, ostensibly, learning new trades.

The ironic thing, of course, is that these use-based systems are designed to be realistic—practice making perfect, and all. Some players and designers are bothered by the

idea that you can learn how to speak Orcish by killing kobolds until you gain a level. A learning-by-doing system makes perfect sense to them.

But in practice, learning-by-doing falls into sort of an uncanny design of game mechanics. Efficient advancement in a use-based system often nudges people to perform odd gameplay that is frequently repetitive as well as not particularly fun. Rather than feeling natural and elegant, the game mechanic feels unnatural and contrived, and worse, draws attention to itself in the process. Learning Orcish by killing kobolds may not be terribly realistic, but at least at no point is the player being asked to do something he didn't want to do anyway.

### REALISM VS. CONSISTENCY

» There's a lot to like about *Gotham Central*. The DC comic was a police procedural set in Gotham city, and tried to describe what it was like to be a detective and have to clean up after Batman and Joker slugging it out amongst the rooftops. The comic ran for 40 issues, earning meager sales but strong critical praise. Those who loved it often cited the series' gritty realism. Which is interesting, given the series still hinges on a man who fights crime dressed as a bat.

A lot of times, people think they want realism when what they really crave is internal consistency within a given universe. *Gotham Central* feels a lot like what happens if you merge the classic *Dark Knight* with gritty TV cop fare like *The Shield*. The goal is to make the rest of Gotham as real as possible, and the end result is a world where Batman is still amazing and mysterious, without becoming silly or ludicrous. He feels possible—even though he's not.

Immersion is the goal. The player should be drawn into your worlds and experiences. Realism is good when it supports immersion, and bad when it gets into the



market that had previously obsessed over realism to an insane degree.

Realism is a choice, both for artists and for designers—but it can also be also a trap, and one that is perilously easy to fall into. In art, chasing realism is expensive—technology can provide incredibly

shoehorn sandbox design principles on almost every genre of gaming. In sandboxes, players are free to go anywhere and tackle content in almost any order, rather than be drawn along a linear game path with unreachable areas blocked off by unrealistic obstacles or invisible





From Software's *DEMON'S SOULS* features gameplay that is derived from a realistic consideration of physical strength and stamina versus weapon weight and encumbrance.

way. For example, most single roomed buildings in games are huge, often with 18 foot ceilings. It's not realistic, but the player rarely notices. On the other hand, he always notices when, in a small room, the camera moves in too close to see or do anything.

Jumping is an interesting place where realism and gaming diverge. Most games that have jumping allow ludicrously high jumps—often a character can leap 6 feet high from a dead stop, because it feels right (see Inner Product November 2009). But recently, some action games—such as *GEARS OF WAR*—have been experimenting with not allowing jumping, since jumping around like a jackrabbit in heat isn't particularly realistic. For the most part, these experiments have been successful—until the player finds an obstacle that he can't jump but could in real life. Even worse, he could clear it by five feet in a game that allows jumping. The obstacle feels unrealistic, and

worse, noticeably so. It's a problem because it breaks immersion.

To some degree, the realism we are bound to is determined not by real life, but by our forerunners. Hit points linger as a concept because most games teach us that you usually hit what you swing at, but fights shouldn't be over instantly. When an NPC tells you to "hurry," he doesn't mean it unless a timer appears on your screen. Rocket launchers aren't just great weapons, they're also solid ways to propel yourself up to a hard-to-reach ledge. But it's not just games—most gun effects in shooters sound more like they do in the movies than they do in real life, because the theatre is where most players learn what automatic gunfire sounds like.

In all these cases, following unrealistic conventions can make the games feel better than taking a more realistic approach that breaks player expectations. Worse, breaking convention can make the game


feel less realistic, even though it is more so.

### THE PLACE OF REALISM

» Designers make concessions to realism all the time, of course. In the real world, it only takes one bullet from an assault rifle to kill a man. Building a breast plate from raw iron doesn't happen in less than 10 seconds. If you get brought near death by the jet of a flamethrower, you aren't likely to be hopping back into battle after a couple of first aid kits. This is before we get to the inherent fantasy of the worlds we build: worlds full of dragons, gangsters, or battlecruisers. And lest we cut out the mundane—short of *THE SIMS*, no games require your characters take bathroom breaks.

But realism can enrich a game as well. An MMO that has crafting can have a much more realistic economy than one that doesn't, even if the mechanics of crafting aren't realistic. An assassin that

trades in poisons feels more real, even if game balance requires that poison be a minor damage over time effect instead of being immediately lethal. Bouts of *MADDEN* that end with scores like 30–27 feel more real, even if it takes five minute quarters to keep the scoring that low.

At the end of the day, players play games escape the real world, so designers shouldn't be such a slave to it. Players are hoping to live a fantasy provided by the game designer. Good games make those fantasies as immersive as possible, but they don't always do that by making them realistic. Sometimes, too much realism gets in the way. 

---

**DAMION SCHUBERT** is the lead combat designer of *STAR WARS: THE OLD REPUBLIC* at BioWare Austin. He has spent nearly a decade working on the design of games, with experience on *MERIDIAN59* and *SHADOWBANE* as well as other virtual worlds. Damian also is responsible for *Zen of Design*, a blog devoted to game design issues.



# DETHRONED

## IS THE AGE OF LOOPS COMING TO AN END?

**BEFORE AMBIENT SOUND,** before dialogue, there was music. Since the early days of game sound, game music has been synonymous with loops. Initially, loops were the only practical solution to multilevel worlds that would have otherwise been filled only with the bleeps of jumps and the bloopings of hit impacts. As the years marched on and carts gave way to discs, loops persisted due to disc size limitations.

25 years on from the release of SUPER MARIO BROS., loops persist in the face of rich surround ambiances, complex interactive dialogue systems, and massive amounts of disc and hard drive space. Perhaps more habitual now than practical, looping music remains an often immediately assumed feature, like pause menus and option sliders.

However, more and more audio teams and music implementers are beginning to question the assumed reign of loops, while looking for different solutions.

### MEET THE NEW BOSS

» As game worlds get larger the weaknesses of loops become glaringly apparent. Larger worlds require either longer loops or more repetitions. As such, listener fatigue is of constant concern to implementation teams. Eventually, loops simply can't be long enough or varied enough to adequately cover a large modern game without diminishing their own impact.

The obvious solution to the problem is actually the

oldest answer to music in games—non-looping music events. Fire up any game of MS. PAC-MAN and you'll find three types of music events: intro stinger, cutscene music, and death stinger. None of these cues loop. More and more these days, game music implementation is beginning to fall back toward that same approach as a means of avoiding listener fatigue.

MMOs like Blizzard's WORLD OF WARCRAFT are simply too gigantic for loops. Anyone who has played WoW will know that spawning into the game initially will always result in some flavor of intro music cue. This cue does not loop, and will eventually simply give way to the ambient sound of the world. At certain intervals afterward, more music will trigger; but again it is all non-looping and always relinquishes the audio focus over to the ambient sound effect systems.

MMOs aren't the only games moving away from loops. The front end menus of console games rarely feature dialogue or ambient effect systems, and as such have been longtime showcases for looping main themes. Now, with an increasing frequency, front end main themes are showing up as longer, non-looping suites of thematic material. DRAGON AGE: ORIGINS, FRACTURE, and UNCHARTED 2: AMONG THIEVES all begin with broad pieces designed to set the mood, which build to a climax, and end in a very traditional linear music

fashion without loop points or variation.

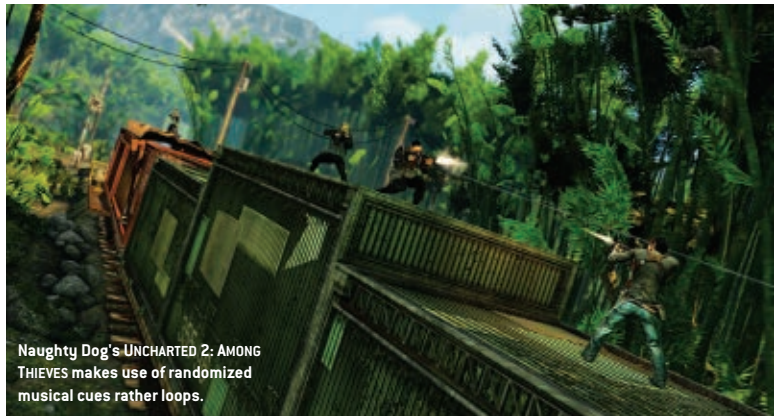
In UNCHARTED 2 Sony's music team experimented with applying a frequently-used technique for looping music to non-looping music with entirely new results. One of the most basic interactive music systems is one in which gameplay is blanketed with looping exploration music. When combat is encountered, the calm exploration loop is replaced with a tense combat loop. In UNCHARTED 2's

sound design work of the level which includes audio-specific "tells" that help the player avoid danger and progress through the level.

Perhaps one of the most telling signs of the changes occurring in music implementation comes from the structure found within Audiokinetic's Wwise music engine. In Wwise, music can't simply be flagged to loop indefinitely. Looping music segments must be defined as children of a Playlist. Playlists can then be looped, but by

systems are still loop-based. Most combat music is still loop-based. Smaller games are still loop-based for all the reasons that loops originally came into such prominence.

Additionally, there's a cultural mindset that needs to change within development teams if we're to ever move drastically away from massive blankets of in-game looping music. A lack of music in-game often gets flagged by QA teams as a bug. Sometimes a simple looping piece of music is



Naughty Dog's UNCHARTED 2: AMONG THIEVES makes use of randomized musical cues rather than loops.

"Locomotion" sequence, the team relied on the ambient sound design of a speeding Nepalese train to fill the role of calm exploration. When the AI attacks or becomes aware of the player's presence, the combat system kicks in. But unlike traditional loop-based systems, UNCHARTED 2 pulls combat cues from a randomized bank of non-looping stingers per combat encounter. The result is tension music that simply comes and goes with combat states, creating a counterpoint to the rich

their very nature are much more flexible, and present to the music implementation team a means of transitioning from one non-looping cue to another with smooth transitions. In short, loops aren't merely the assumed default option any longer and the implementer must choose at the outset to use loops or approach the problem from a different direction.

### SAME AS THE OLD BOSS

» Looping music isn't going away any time soon. The vast majority of interactive music

used to cover over what might otherwise be a boring segment of gameplay. And sometimes, this is a great move. Other times, however, music becomes a fatiguing crutch that we don't need any longer as an industry. Games, technology, and audiences are all ready for us to experiment more. The only limitations are our own infinitely looping habits. 🎧

**JESSE HARLIN** has been composing music for games since 1999. He is currently the staff composer for LucasArts.



THINK SERVICES  
GAME GROUP

NETWORK OF SITES

[www.gdmag.com](http://www.gdmag.com)  
**game**developer



**GAMASUTRA**

The Art & Business of Making Games

[www.gamasutra.com](http://www.gamasutra.com)

 **GAME  
CAREERGUIDE**


[www.gamecareerguide.com](http://www.gamecareerguide.com)

 **FG FINGER  
GAMING**

[www.fingergaming.com](http://www.fingergaming.com)

**GAME SET WATCH**

[www.gamesetwatch.com](http://www.gamesetwatch.com)

**GAMERBYTES** 

[www.gamerbytes.com](http://www.gamerbytes.com)

**WORLDS IN  
MOTION.BIZ**

[www.worldsinmotion.biz](http://www.worldsinmotion.biz)

**indiegames.com**

[www.indiegames.com](http://www.indiegames.com)

**SERIOUS  
GAMES SOURCE**

[www.seriousgamesource.com](http://www.seriousgamesource.com)

**THINK  
SERVICES**  
A DIVISION OF UNITED BUSINESS MEDIA, LLC



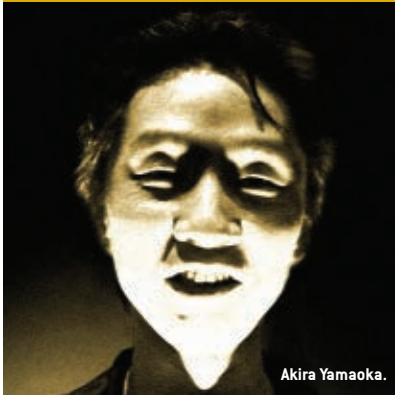
## konami sound team exodus

### MICROSOFT'S INTERNAL HALO GAME DEVELOPMENT TEAM 343 INDUSTRIES HAS HIRED SOTARO

Tojima, a longtime Konami employee who most recently served as sound director on Kojima Productions' METAL GEAR SOLID 4: GUNS OF THE PATRIOTS.

Tojima will take the similarly-named role of audio director at 343 Industries, according to a short post made to the studio's official Twitter account.

The 343/Kojima link likely comes from Ryan Payton, a well-known associate producer at Kojima



Akira Yamaoka.

Productions who left that studio after METAL GEAR SOLID 4 shipped to take a creative director role on an unannounced HALO game, which was being developed by a pre-343 Microsoft Game Studios team.

This news comes after SILENT HILL composer Akira Yamaoka left the Konami fold (word is he may be starting his own company—that will be one to watch for). A scant two months prior, recent CASTLEVANIA series musician Michiru Yamane left the company as well.

What's going on at Konami, you may ask? This is more common than it appears, in Japan at least. Square Enix bled its higher-level composers when FINAL FANTASY songsmith Nobuo Uematsu left to form his own company—but he

still composes music for the company to this day. Only now, he's free to compose for others as well. Expect more of that from Yamaoka and Yamane.

## who went where?

» Nexon America, online publishing arm of Nexon Corp. brings former parent company CEO Won Il Sue (2004-2005) to the U.S. as Biz Dev VP. Sue comes off a stint at rival company Neowiz.



Won Il Sue.

» Publisher/developer True Games gains former Electronic Arts exec Mick Giles as CTO—he

was most recently senior director of online operations for North America and Asia and senior director of technology for Worldwide Studios at EA.

» Outsourcing and development group Blade Games grabs former producer and general manager of EA Pogo Shanghai Justin Weng as head of product development, and a new senior art director in Tony Zander, most recently of Microsoft's FLIGHT SIMULATOR team. Zander is an 18 year art vet, and will lead the cross-continental company's outsourcing and internal art efforts.

» Berlin-based social games developer Wooga added to its board of directors one Roberto Bonanzinga, a partner at the European VC firm Balderton Capital.

» Production software maker Hansoft has hired EA Sports veteran Chris Wynn as senior production expert. He began as an artist, but moved into production roles over his 12 years in the industry.

» Playdom, social game dev, hires new CFO Christa Quarles, previously an analyst at investment banking firm Thomas Weisel Partners.

» Online publisher and developer Trion has hired two veterans in David Reid and Scott Hartsman. New senior vice president of publishing Reid was NCsoft West's publishing president, while Hartsman, now Trion's chief creative officer and Redwood Shores Studio general manager, previously spent seven years at SOE working on the EVERQUEST games.



portal  
designer  
kim swift  
joins  
airtight  
games

**KIM SWIFT, A DESIGNER OF VALVE'S ACCLAIMED FIRST-person platformer PORTAL, has left the HALF-LIFE creator to build up a new team at nearby DARK VOID developer Airtight Games. The news was revealed in an announcement by Airtight. The company says Swift will hold the title of project lead at the studio, and will "build and lead a team in the development of games aimed at a more diverse audience."**

No details about the nature of her first project were given, but Airtight Games president Jim Deal said her hiring "represents a strategic move by Airtight Games into new and broader gaming markets." Game Developer spoke with Swift to get a little insight into her company move.

### Why did you choose Airtight?

**Kim Swift:** *I love the fact Airtight is a small, independent developer with some really great people working there. I haven't been here for very long, but everyone I've met has an awesome sense of humor and is super talented. I also had a couple friends who worked at Airtight already and I really valued the opportunity to work with them.*

### What do you hope to accomplish that's different there?

**KS:** *I'm definitely interested in creating games that are more accessible, and Airtight has similar views. I think the more video games become a mainstream medium, people are going to want to play games with their spouses or children, and I think the marketplace could use a few more games with a broader appeal.*

### Getting more power seems like a good thing, since you're leading a team, but won't a lot of the people be older than you? Is that weird at all?

**KS:** *I don't think it's strange at all, at the end of the day we're all trying to put together the best game possible. Everyone has to pitch in and work together as a cohesive team and age shouldn't factor in at all. We're all just trying to do our jobs and hopefully have some fun at the same time.*

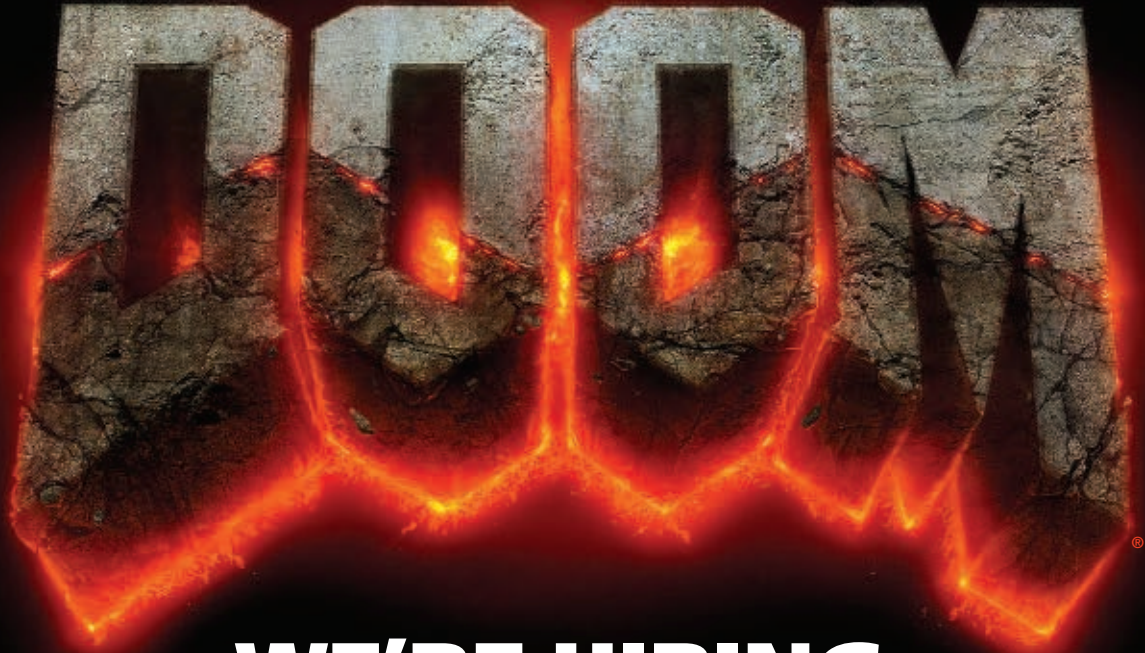
### Do you feel confident that PORTAL could continue at Valve in your absence?

**KS:** *Well, that's definitely up to Valve, but I'm sure PORTAL will be just fine.*

### It's early days at Airtight, but having been essentially "trained" at Valve, how do you think it will be moving to another production style?

**KS:** *I've been given the opportunity to run a team of really great developers much in the same way that teams are run at Valve. Everyone who's working on the game gets to pitch in on the design decision-making, which I think not only is great for morale but is more efficient as well. Game design is essentially problem solving, and getting different disciplines to look at the same problem together will help ensure that the best solution is found.*

**IF YOU'RE LOOKING  
TO WORK ON THE  
NEXT BIG THING**



**WE'RE HIRING.**

**id Software is now hiring for:**

**Artists  
Animators  
Designers  
Programmers  
Producers**

**Scripters  
Marketing/PR  
QA  
IT**

[www.idsoftware.com/makedoom](http://www.idsoftware.com/makedoom)





## EDUCATED PLAY!

# student game profile: AZ66

## EXPLORING NEW IDEAS IN GAME CONTROL

AZ66 is an intriguing UNREAL mod created by students at France's ENJMIN (National School of Video Game and Interactive Media) that combines real-time data generated by the player's heart rate with a devious "capability test" that has no win condition. Set in a dystopian world where citizens are given logic and reflex tests to determine their usefulness to society, AZ66 evokes a sinister mood and an elevated sense of panic as the player's heart rate has an increasingly disruptive effect on their perception of the game. We talked with AZ66's designer Stéphanie Mader to find out how the game (literally) gets under the player's skin.

### Jeffrey Fleming: Can you tell us a bit about the history of the project?

**Stéphanie Mader:** We are students at ENJMIN, the only public institute in Europe to deliver a Master's degree in games and interactive digital media. As our school is focused on the specialties of game design, graphic design, sound design, programming, project managing, and ergonomics, a big part of the program is to teach us teamwork, with responsibilities

*divided according to each member's specialty. During the first school year, we had to make several video games as a team.*

### JF: How did the team get together and divide responsibilities?

**SM:** Before starting the project, some members of the AZ66 team had already worked together. So, when we came up with the idea of doing a video game using heart sensors, part of the team reformed and some new members joined, in order to have a full development team of six students from each specialty area.

### JF: How much time did it take to complete AZ66?

**SM:** For this project we had three months, but we had classes and exams too. On top of that, each member was working on other projects at the same time. Planning the teamwork was a true puzzle.

### JF: How did you approach AZ66's game design?

**SM:** Due to the heart sensors, the game had some concept challenges. It was all about bluffing the player and getting him nervous. To accentuate our game design intention to its maximum, AZ66 includes a bomb that can't



The AZ66 team.

*be defused, never-ending targets that pop up, and a labyrinth with no exit. Dark and dirty visual art, disturbing sounds, and the constant presence of voices all work to distract the player.*

### JF: On the technology side, what was the most challenging?

**SM:** We had to learn to work with Unreal Engine, its architecture, its tools, its pipeline, and its script language. As a result, programming and asset integration took most of our time. At the same time, our programmer had to deal with getting the sensor data into the game, while our usability expert worked on the player's comfort when wearing the sensors. But Unreal Engine was also what made our project achievable!

### JF: I'm curious about the biofeedback element in AZ66. How did you measure heart response in the

### player and how did you connect it to gameplay?

**SM:** We were lucky, because we had strong support from Jérôme Dupire who works at CÉDRIC, the CNAM French state institute in computer research and development. Jérôme's team created a new version of their PLUG Project with heart sensors, skin response, and temperature, a total of five sensors connectable to the player. In fact, we only used data coming from the heart sensors, due to time constraints. We used the player's variations in heart rhythm to give real-time feedback on his emotional state.

*But remember, it is all about bluffing! So the heart rate data didn't really affect gameplay directly, instead it affected the player's perception of the game. We wanted the player to think that his heart response was important by making it audible during the whole*

*game, so he could hear his loss of control. In response to his heart acceleration, we applied graphics filters like motion blur to the screen. To complete our deception, we added some contextual feedback. For example, in the first room, the graphic and sound of the bomb timer becomes strange. In the same way, the speed of the cube that follows behind the player in the labyrinth is directly connected to his heartbeat.*

### AZ66 TEAM

#### Project Manager:

Stéphan Froment

#### Game and Level Designer:

Stéphanie Mader

#### Programmer:

Antoine Sarafian

#### Environment Artist:

Delphine Soriano

#### Sound Designer:

Aymeric Schwartz

#### Ergonomics:

Mélanie Ginibre

#### Voices:

Laure Nowak and Aymeric Schwartz

## AZ66 links

**AZ66** <http://az66.interaction-project.net>

**ENJMIN** [www.enjmin.com](http://www.enjmin.com)

**PLUG Project** <http://interaction3d.fr>

# geeked at birth



Learn

Bachelor of Science > Game Programming  
Bachelor of Arts > Game Art and Animation, Game Design, Serious Game and Simulation  
Master of Science > Game Production and Management

## ⚠ CLUSTERGEEK WITH CAUTION



LEARN, EXPERIENCE AND INNOVATE WITH THE FOLLOWING DEGREES: Advancing Computer Science, Artificial Life Programming, Digital Media, Digital Video, Enterprise Software Development, Network Engineering, Network Security, Open Source Technologies, Robotics and Embedded Systems, Technology Forensics, Virtual Modeling and Design, Web and Social Media Technologies

You can talk the talk.  
Can you walk the walk?  
Please geek responsibly.



[www.gamedegree.com](http://www.gamedegree.com)



## Become a technical artist

Bachelor's degree in Game Development  
Bachelor's degree in 3D Arts

Visit us at GDC 2010

[www.digitalartsandentertainment.com](http://www.digitalartsandentertainment.com)



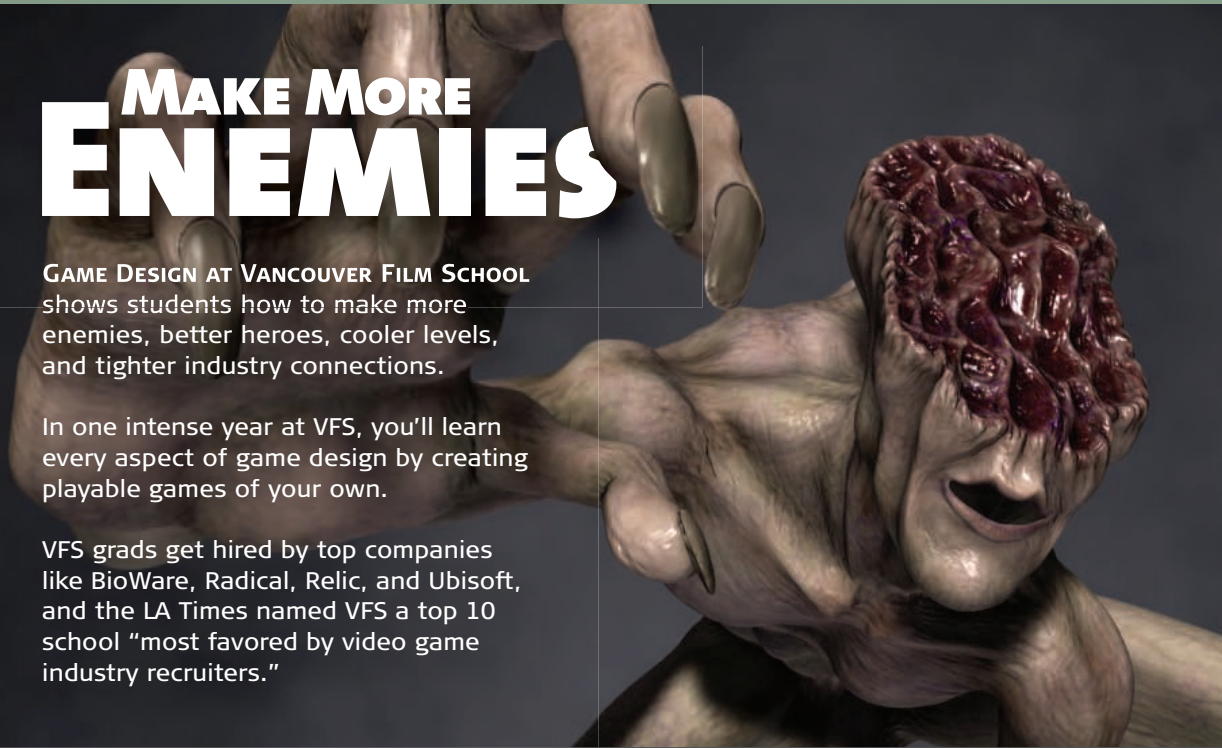
**howest** University College West Flanders  
Belgium | Europe  
[www.howest.be](http://www.howest.be)

# MAKE MORE ENEMIES

**GAME DESIGN AT VANCOUVER FILM SCHOOL** shows students how to make more enemies, better heroes, cooler levels, and tighter industry connections.

In one intense year at VFS, you'll learn every aspect of game design by creating playable games of your own.

VFS grads get hired by top companies like BioWare, Radical, Relic, and Ubisoft, and the LA Times named VFS a top 10 school "most favored by video game industry recruiters."



**VFS**

[vfs.com/enemies](http://vfs.com/enemies)

VFS student work by  
Moo Won Kim



# TURN YOUR PASSION FOR GAMING INTO A CAREER

## Game Art

Bachelor's Degree Program  
Campus & Online

## Game Design

Master's Degree Program  
Campus

## Game Development

Bachelor's Degree Program  
Campus

## Game Design

Bachelor's Degree Program  
Online



## Campus Degrees

### Master's

Entertainment Business  
Game Design

### Bachelor's

Computer Animation  
Digital Arts & Design  
Entertainment Business  
Film  
Game Art  
Game Development  
Music Business  
Recording Arts  
Show Production  
Web Design & Development

### Associate's

Graphic Design

## Online Degrees

### Master's

Education Media Design & Technology  
Entertainment Business  
Entertainment Business: *with a Sports Management Elective Track*  
Internet Marketing  
Media Design

### Bachelor's

Computer Animation  
Entertainment Business  
Game Art  
Game Design  
Graphic Design  
Internet Marketing  
Music Business  
Web Design & Development



**FULL SAIL**  
UNIVERSITY

800.226.7625 • 3300 University Boulevard • Winter Park, FL 32792

Financial aid available to those who qualify • Career development assistance  
Accredited University, ACCSCT

[fullsail.edu](http://fullsail.edu)

## ADVERTISER INDEX

COMPANY NAME	PAGE	COMPANY NAME	PAGE
Autodesk.....	13, C3	RAD Game Tools .....	C4
Bethesda Softworks .....	51	Seapine Software .....	19
Epic Games .....	14, 26	University College of West Flanders .....	54
Full Sail Real World Education .....	55	University of Advancing Technology.....	53
Havok.....	C2	Vancouver Film School .....	54
Perforce Software.....	3		

*Game Developer* (ISSN 1073-922X) is published monthly by United Business Media LLC, 600 Harrison St., 6th Fl., San Francisco, CA 94107, (415) 947-6000. Please direct advertising and editorial inquiries to this address. Canadian Registered for GST as United Business Media LLC, GST No. R13288078, Customer No. 2116057, Agreement No. 40011901. **SUBSCRIPTION RATES:** Subscription rate for the U.S. is \$49.95 for twelve issues. Countries outside the U.S. must be prepaid in U.S. funds drawn on a U.S. bank or via credit card. Canada/Mexico: \$69.95; all other countries: \$99.95 (issues shipped via air delivery). Periodical postage paid at San Francisco, CA and additional mailing offices. **POSTMASTER:** Send address changes to *Game Developer*, P.O. Box 1274, Skokie, IL 60076-8274. **CUSTOMER SERVICE:** For subscription orders and changes of address, call toll-free in the U.S. (800) 250-2429 or fax (847) 647-5972. All other countries call (1) (847) 647-5928 or fax (1) (847) 647-5972. Send payments to *Game Developer*, P.O. Box 1274, Skokie, IL 60076-8274. For back issues write to *Game Developer*, 4601 W. 6th St. Suite B, Lawrence, KS 66049. Call toll-free in the U.S./Canada (800) 444-4881 or fax (785) 838-7566. All other countries call (1) (785) 841-1631 or fax (1) (785) 841-2624. Please remember to indicate *Game Developer* on any correspondence. All content, copyright *Game Developer* magazine/United Business Media LLC, unless otherwise indicated. Don't steal any of it.



# DEAR MR. WASTELAND

WELCOME TO OUR "Q&A" DEPARTMENT

## IPHONE RICHES

**Q.** A guy I know quit his job after an iPhone game he made in about ninety seconds sold like crazy for some reason. Last I heard, he's been tooling around in a brand new exotic sports car and has hired a full-time entourage. How can I get in on that action?

—Avaricious in Austin

**A.** The secret to a successful iPhone game is to remember that the App Store is more of a spiritual system than a platform. With your game ready to go, consult your astrologer for auspicious release windows, then sacrifice an animal in the usual manner. Be on guard for malign forces who will attempt to harm your game with their cruel magicks. Whatever you do, stay the path: Apple's munificence is only rewarded to its most faithful and devout. Go now, seeker, and may your App be Featured!



are pretty much all you need to get started on the road to indie stardom, after all. You don't even have to drink any Mountain Dew to win the "Best Indie Game Fueled by Mountain Dew" category at the Spike Video Game Awards—talk about easy!

## TALKING TO THE FAMILY

**Q.** Visiting with my relatives over the holidays, I was reminded how embarrassed I am to be a video game designer. My uncle is a medical doctor who developed a new non-invasive surgical technique and my cousin just came back from a stint in the Peace Corps where he helped build a clean water supply for poor villages. Compared to stuff like that, COMBAT ZOMBIE EXPLODER's triumphant 91 Metacritic score just doesn't seem as awesome as it used to be ... how can I measure up?

—Self-Conscious in Boston

**A.** Games are never going to get out of the cultural ghetto if you keep having that attitude. Next time you're at dinner with your sanctimonious relatives, go on the attack, and remember the invincible trump card that we use in all situations to remind people that we matter: the game industry makes a lot of money. For example, point out that the Peace Corps' entire 2010 budget request of \$373 million was totally dwarfed by MODERN WARFARE 2's first five day sales of \$550 million. That should shut up the do-gooders for a bit.

## MIDDLEWARE WOES

**Q.** Look, I know it's the Front Line Awards issue and all, but why is game

middleware still so frustrating so often? Stuff like out-of-date documentation and flaky technical support makes me want to kill something!

—A Furious Dude in Chicago

**A.** Thank you for contacting us about your problems with game middleware. First, I'd like to direct you to our forums, where many users are often able to answer each other's basic questions. If you believe you have found a bug, please search our bug database to make sure it does not already exist, then enter it along with precise steps to reproduce as well as your company's proprietary code surrounding its implementation. Finally, if you are requesting new features, please check our road map to see if the feature is already mentioned. If it is, please remember that a road map is not really a plan as such, but more of a catalogue of various possibilities. Thank you, and have a nice day.

## COOKING AT WORK

**Q.** It's way past midnight and I'm incredibly hungry, but I don't want to go out to get food and all the producers who know how to order pizza have left already. Isn't there anything I could eat so that I don't die?

—Famished in Foster City

**A.** Many people don't know that you can make a delicious meal out of ingredients commonly found around the typical video game studio. Start with a bagel left over from Bagel Friday. Spread some peanut butter on top and crumble Goldfish or Cheese Nips over the whole thing (Doritos can



be substituted if there are no crackers). Warm with the heat from an early-model PS3 development kit, and serve with surplus ketchup or soy sauce packets to taste.

## OUTSOURCED?

**Q.** Recently, my bosses have been talking about making art bibles and putting our visual direction into easily digestible documents that can be sent to potential "partners." As an artist, I think I know what that means: I'm about to be outsourced. Help!

—I'm So Fired in Kirkland

**A.** You're in a tough situation there for sure. How can one person stand up against the tidal wave of global economics? I think your best strategy to defend against this is to quickly become a surly, wearied veteran who criticizes everything around you. That's right—just keep talking about how old you are and how long you've been in the industry. The insecure managers in charge of the outsourcing effort will be frightened by you and will ironically decide to keep you on for your "experience" while letting go of the talented junior artists who were doing most of the work for less pay. Good luck! 🍀

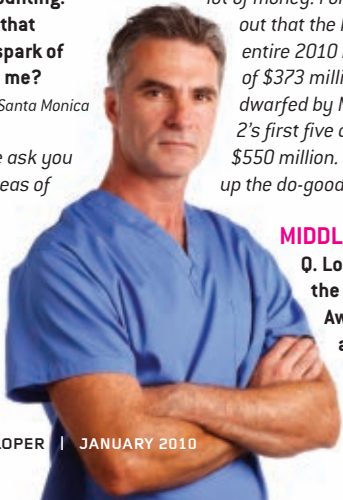


## CREATIVE CRISIS

**Q.** I'm really depressed lately. Why does our industry keep making the same thing over and over? I am getting so frustrated with this business and am considering going into something a little more creative, like accounting. Is there anything that might revive the spark of inspiration inside me?

—Sad in Santa Monica

**A.** Well, Sad, let me ask you this: if you have ideas of your own that you want to do, why not leave? A MacBook, a fixie and a favorite non-Starbucks coffee shop



# Autodesk Games Insight

The latest scoop from Autodesk Media & Entertainment



## Autodesk® Kynapse®

With the demands of more realism in games, how can artificial intelligence help add to the sense of immersion for players? We find out how Autodesk Kynapse middleware can be used in a common scenario to provide realistic AI for games.

*Autodesk Kynapse in action: In this scenario, three bodyguards automatically provide cover for a user-controlled player.*

### Heightening Realism with In-Game Artificial Intelligence

With the technological power of today's consoles and PC's, gamers are demanding more realism than ever. In the challenge to create truly immersive gameplay, artificial intelligence is paramount to giving users an immersive experience. In this article, we look at a common, but challenging game scenario and how the Autodesk® Kynapse® artificial intelligence middleware can help solve this challenge through *3D spatial awareness, 3D path-finding and team coordination*.

In a combat game scenario, we have a player controlled character, and three computer-controlled bodyguards that follow the player and protect it. Wherever the player goes in the level, the bodyguards will automatically identify key areas where an enemy could shoot at the player. The bodyguards will use this information and move into the appropriate position to protect the player. This scenario would be used in games where a player character has computer-controlled team members that need to move through a level with the player and assist.

The AI that drives this kind of behavior is achieved through the use of Kynapse. Spatial awareness libraries help each entity identify in real-time the key topological zones where an enemy could hide and shoot from. This includes

windows, street corners and access ways. These key zones can be automatically generated through the Kynapse toolset, or manually tagged by level designers. What is interesting about this scenario is the realism of the solution – when the player is positioned against a wall, none of the bodyguards will face the wall because no threat can come from the wall.

To move the characters into their appropriate positions, we use the Kynapse 3D path-finding library. This enables each character to find their way to their target positions in the level, while dynamically avoiding objects and other characters. In this scenario, characters only have to move a short distance, but the Kynapse library can also cater for very large maps such as those used in role-playing games. In such cases, Kynapse can utilize hierarchical path-finding technology, which provides an



*Kynapse spatial awareness in action: the bodyguards do not face the wall as no threat can come from it.*

elegant and highly optimized solution for helping characters find their way across larger worlds.

Finally, the “bodyguards” scenario uses team coordination to ensure that each character doesn't go to the same spot. Through the use of the Kynapse libraries, each character can share information, just like a real team would. The result is that each character intelligently moves into a position that could protect the player in the most sensible way, which heightens reality and gives players an immersive experience of being protected by comrades on the battle-field.

This is just one scenario that shows how Kynapse can be used to solve a complex AI challenge in games. In such a case, developing your own AI solution might be prohibitively expensive due to the cost of research, development, testing and implementation. With Kynapse, the solution is “off-the-shelf” and ready to integrate into your game.

To see Kynapse in action (including the above scenario and others), we invite you to watch the Introduction to Kynapse video at [www.autodesk.com/kynapse](http://www.autodesk.com/kynapse).

For more information about Kynapse and Autodesk Middleware, please contact us at [middleware@autodesk.com](mailto:middleware@autodesk.com).



WHAT'S IT LIKE USING

# RAD

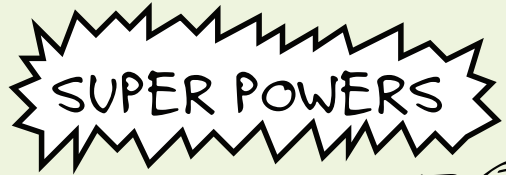
GAME TOOLS ?



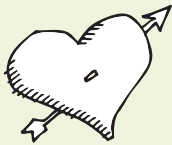
.....



It feels like you have



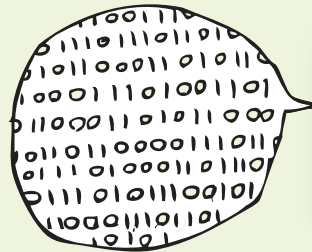
when you use Bink - our amazing, super *FAST* video and audio codec - all in a simple, clean API.



You'll fall in LOVE with Granny, our run-time dynamic

## 3D ANIMATION SYSTEM

with AMAZING content exporters.



And with Miles, our multichannel



sound system, you get the world's *FASTEST*

## MP3 and Ogg DECODERS

+ way cool *new* high-level audio tools.

.....

# IN A WORD, USING OUR TOOLS IS rad!



[www.radgametools.com](http://www.radgametools.com)  
(425) 893-4300